置身钉内

楔：钉钉是一只雨燕

钉钉的动物园形象钉三多，是一只尖尾雨燕，世界上平飞最快的鸟类之一。它最特殊的地方在于，可以吃喝、睡眠、交配通通在空中完成；每年从非洲飞经印度和东南亚，回中国北方繁殖，最多可以连续飞行三百多天不落地。

> 尖尾雨燕的拉丁文是apus apus，「pus」是「脚」的词根，「a」是「没有，否定」的词缀，所以雨燕是字面意义上的「没有脚的鸟」。

我在 2025 年 6 月 入职钉钉，入职即加入了作为核心保密项目的「O项目」。后来我才知道，「O」代表「ONE」。说起来我应该是ONE最晚进入的核心 PD，同时是最后一个留下并送走 ONE 的。我参与了从 0 到 1 的迭代，接触了从项目成型开始，到项目暮年运营期所有阶段，所有核心决策。也目送走了许多同事。

在钉钉飞行了 300 多天，将满一年，最近也到了重新踩回地面的，离开的节点。

ONE是一个怎样的项目

ONE 的生命周期从 2025 年的 4 月份开始孕育，8 月 25 日发布会首次公开，DAU 巅峰稳定在 300 万左右，是无招回归后第一个主推的号称 AI 原生的项目 。



影响 ONE 一系列产品决策的，主要是产品设计团队。ONE 的第一个一号位曾经是设计中心的负责人，这决定了 ONE 是一个设计基因极重、参与决策浓度很高的项目。

据我所知的，参与过 ONE 的产品设计是不少的，但 ONE 是一个流动性极大的项目。我来的第二周，我的设计 leader（也是第一个一号位）就离开了，第四周，联系并推荐我进组的师兄也被调离去了其他部门。
在 ONE 超过 3 个月的产品只有 3 个人，我是其中一个。

本文是一篇怎样的内容

本文将夹叙夹议地回首，这样一款曾经站在战略级风口位置的 AI 产品的开发往事。

我想记录一个 AI 产品从立项、发布、共创到收缩的过程中，理想如何被翻译成目标，目标如何被拆成需求，需求如何在组织里变形、落地，如何进入真实的用户现场——而退远镜头的景别，这一切究竟如何在钉钉、在时代的背景下发生。

「凡历术在于常数，而不在于变行。」编订历法，需要归纳规律性的常数，而不在于一味记载关注短期变化。感谢钉钉的高强度和快产品节奏，我在十个月内就完整体验了一款大用户量级的AI产品的生老病死。这实在是非常宝贵的经验。因此，这篇文档既想记录我这三百多天亲历的「变行」，也想尽可能凝练出一些在反复实践、受挫和复盘之后得到的「常数」。

当然，这里的「常数」可能不是把一个项目的成败归纳成几条规整、漂亮、让读者有充分获得感的原则。可能它只是一场现场经验留下的残余判断：在高速变化的 AI 产品开发中，哪些问题看似每次不同，其实反复出现；哪些冲突看似来自具体的人和事，其实来自组织、技术、商业与用户之间长期存在的结构性张力；哪些选择当时像偶然，事后看却几乎必然。

这份记录必不可能完全客观，我无法、也不愿意假装站在一个无菌的观察位上。很多判断来自我亲历的会议、争论、上线、复盘和失落，也来自我作为最后留下的 PD，对一款产品从热烈理想走向收缩的复杂感受。但也正因如此，它或许能保留一些正式复盘里不容易出现的东西：真实的犹豫、判断的代价、组织的惯性，以及一个产品人在现场里倍速积累的经验。

总之，希望这份记录能给所有正在、或希望从事 AI 产品开发的同事、同学与同道，提供一汪小小的经验池。若你能借此印证一些观点，或避开一些坑，那我便觉得略尽绵力，已经值得。

总之，请汉娜阿伦特和茨威格上我的身，请你和我一起翻开属于 ONE 的，昨日的世界。

FBI Warning （补充说明）

本文成文较长，7.5 万字左右，三易其稿，是最近一两周工作间隙 + 紧急大熬夜写的；所以如略有前言不搭后语之处，敬请包涵雅正。

发心第一

发心是行为的内在动机与根本誓愿，是一切工作、生活、实践、修行的起点、方向和动力源泉。一个理性善良人的行为的发心，往往在帮助世界、物质回报和成就自我之间逡巡。在不同阶段，三种发心会有不同的优先级，促使人做出不同决策。

产品也有产品的发心。产品的发心就是它的发起人最原始的出发点，主要有以下几类：

- 解决某个尚未被解决的具体问题，
- 提升解决某个问题的效率，
- 服务好某个具体的人群，
- 推广某种理念，
- 销售某种资源，

等等。

举几个具体的例子：

1. 「淘宝」，发心是「提升解决商家对接客户销售地问题的效率」，也就是「让天下没有难做的生意」。
2. 「安缦酒店」，发心是「服务好度假的高净值人群」。
3. 「番茄钟」类产品，发心是「推广25min专注+5min休息的工作节奏的理念」。

上述列出的发心类型中，越靠前的越有可能诞生有历史价值的产品，因为发心越纯粹。虽然和人一样，一个产品也会混合以上多种发心，但大部分情况下，好产品只有一个主发心。大道至简，这也和许多投资人会提倡的「一句话说清产品价值」异曲同工。

当一个产品的发心又多又没有主次的时候，就会成为一个贪心而焦虑的产品。

贪心是七罪之一。什么都想要，容易什么都得不到。这一点在下文也会阐述更多。

环境背景

外部环境：2025 年的风向

2025 年中，整个大环境的 AI 产品的叙事正在换挡。

前一阶段，大家还在惊叹模型会聊天、会总结、会搜索。到了这一年，行业更关心 AI 能怎样调用工具、完成任务、进入真实工作流。

5 月， Google I/O 2025举办，Google 在 Search 模块里强调 AI Mode，也在 Cloud 侧推出和增强了 Agent Development Kit、Agent Engine UI、Agent2Agent 等 agent 相关能力。它的重心已经从降低幻觉、回答问题的叙事，走向让模型进入更复杂的应用生态和工作流程。

7 月，OpenAI 发布 ChatGPT agent，强调 ChatGPT 可以用自己的工具和计算机完成研究、预订、制作幻灯片等任务。

9 月，Anthropic 发布 Claude Sonnet 4.5，重点也落在 complex agents 和 computer use 上。

钉钉有组织关系，有消息、日程、待办、会议、文档、审批等一系列产品。AI 如果要进入真实工作现场，钉钉这样的办公软件天然比一个单独的聊天机器人能满足用户需求。

这也是我当时被钉钉吸引的原因之一。银泰的经历让我切实意识到， LLM 在缺少 context 和不服务用户刚需时，有多鸡肋。

当然，优势也伴随着麻烦。钉钉不是白纸，它有多年积下来的产品逻辑、权限系统、端侧差异、多组织问题、客户定制和用户习惯。AI 要在这里做事，必须穿过旧系统的技术债。它站在一个很有吸引力的风口，但是取景再往远退一舍，就会发现，这个风口正处在一片难以改造的旧城中央。

内部环境：无招回归

另外，无招的回归本身也是一个非常奇特的变量。

一个人绕了一大圈，又回到自己曾经亲手点燃火种的迦密山。

来往失败后，无招带着几个人退到湖畔花园，从一片败局里做出钉钉。那几年，DING、已读未读、企业通讯录、审批，把钉钉从微信的阴影里硬生生拧将出来。那些功能后来被很多人批评为强管理、强控制，但在当

年，它们确实回答了企业里最朴素也最焦虑的问题：我说的话，对方到底看见没有；我交代的事，到底有没有往前走。

更具体一些，解决了管理者的问题，解决了一个企业里愿意付钱的人的问题。

一个产品经理最难摆脱的，往往不是失败，而是成功。因为失败会留下伤口，而成功会留下手感。

钉钉早年的胜利，给无招留下了一套很深的身体记忆：站在发信人一侧，替组织争取确定性，用强触达把事情往前推。这套记忆后来也进入了 ONE。为什么卡片里的消息一定要算已读，为什么系统要主动把事推到用户面前，为什么钉钉总忍不住替组织催促人——很多答案，都可以回到这里。

但 2025 年 4 月的无招，已经不是 2014 年湖畔花园里的无招。

他曾经离开钉钉，被调离自己一手做大的产品；也曾经离开阿里，去日本做了「两氢一氧」。那个项目没有跑出来。一个人最耀眼的战功留在过去，新的创业又未能证明自己，这时候再蒙旧主召回，很难只是一次普通任命。更何况召他回来的人是在他职业生涯几个关键节点上反复出现的学长，像命中有一根透明而柔韧的鱼线，把他从阿里第一位实习生、淘宝搜索、来往、钉钉，又牵回钉钉。

创立钉钉，被迫远走重回再挂帅。

此人此景，似曾相识燕归来。

Stay Hungry, Stay Foolish。无招的钉钉签名用它，钉钉的文化衫印它，在内部外部讲话，他也在寻找精神偶像的寄托时，每每提到那个男人——乔布斯。

乔布斯讲「stay hungry, stay foolish」的出处，是 2005 年斯坦福毕业典礼演讲。他讲到自己的辍学，讲到去印度学禅修、学字体课，讲被苹果赶走，讲感到死亡临近；最后说起《全球概览》封底上的一张乡间小路照片，底下印着 Stay Hungry, Stay Foolish。他把这句话送给毕业生，也像是在给自己一生盖印：不要太快餍足，不要太早聪明，保持不满，保持探索。

无招身上也有这种叙事的诱惑。创立钉钉，又离开钉钉；出去折腾一圈，再被召回旧地。一个人骤然发现命运之书被风吹开到和偶像相似的一页，很难不教人动心、疑心这是命运的召唤。

不过乔布斯离开苹果以后，做了 NeXT 和 Pixar。NeXT 商业上并不耀眼，却带回了苹果后来操作系统的骨架；Pixar 则已在外部世界证明了自己。

无招回钉钉时，手里只有商业结果存疑的 HHO。它更像一次不甘心的出走。证明他仍然想做事，出去转了一圈却没有带回更多筹码。背上背着的还是已经长出半刃铜锈的朴刀。

身后是淘天的旧梦，来往的败绩，钉钉的胜利，HHO 的空响；眼前是 AI 重新洗牌的机会，时代重新开局的诱惑。一个人半生的伤口和功业，忽然都指向同一个方向，便很容易生出一种近乎宿命的确信。

ONE 后来的许多气味，都可以从这里闻到：它急着做成新入口，急着证明钉钉没有老，急着把 AI 落到消息、日程、审批和组织关系里；同时，它也带着旧钉钉的手感，带着发信人立场，带着强触达，带着把事情往前推的固执。

也正因为如此，ONE 这个项目本身，对他有一种特殊的吸引力。

我忍不住现象，在 25 年初，也许还在日本，也许他在见吴妈前，也在镜前反复练习和自问：当年我用钉钉把企业协作从微信嘴里抢出一口事业来；十年后，AI 要重写办公软件，我怎么带领团队，再搞一波大的?

镜子里的人脸上写满野心，也写满迫切和焦虑。后来的很多设计和争议，皆源于斯。

个人背景

最美逆行者

我在去年六月来到钉钉。

在那个节点，很多朋友开玩笑说我是「最美逆行者」。原因是，四月初无招回归后，雷厉风行地举行一系列措施：工时调整——提前到9点上班、开固定晨会晚会、午休缩短、周末单休，再兼全员Python考试，节假福利削减，部分职级以上薪酬调整，还有要求不私下加微信、需要说「对不起我只用钉钉」等等，组织整体在人口净流出。

岔开一句：这个人口净流出的数据结果，可能也是组织需要的，尽管流出的原因值得斟酌。伴随着流出数据的是大家对新入职钉钉的同事的同情和打趣，是就业者对钉钉的规避（并非虚言，君见各社媒劝退贴）（不过删帖公关也快，能不能见，要看手速），是钉钉像孔乙己一样走进咸亨酒店，空气里就弥漫着快活的气氛。

收回——总之，现在回想起来，那时种种debuff，重峦叠嶂，但由于更在意具体做什么事情，以至于我对这一切毫不敏感，在当时有其他明显更好条件的offer的情况下，跳上轻舟，平薪平职来了钉钉。

前情结算

在钉钉之前，我在银泰工作了一年多。也是做AI产品，做了银泰喵街app里的AI导购「喵街答答」和一些其他相关的AI服务——可以理解为银泰版淘宝问问（现 「AI 万能搜」）。为什么来钉钉呢？一方面，彼时正值银泰出售，新东家是传统企业，对AI和产研的投入逐渐不明；另一方面，银泰的工作实践也让我深刻意识到一件事，就是LLM有效服务成立的条件有两个，而银泰彼时的环境条件都不那么具备：

1. 【必要条件】足够的数据基建和context；
2. 【充分条件】最好用户想要消费的对象本身，就是LLM生产的产物。

智能是平权的，但是 context 是不平权的。

有context，才能判断用户的背景和偏好，才能提供用户想消费的商品/服务/内容。

其次，用户在电商零售场合最终想要获得的是「更好」或「更有性价比」的商品，LLM只是一个路径，在对比多种路径的性价比和给用户带来的损耗/收益比的平衡上，就会非常如履薄冰。电商的本质是「商」，是商品，是把渠道对接到销售——AI可以加速、或者优化这个本质的体验，但不会取代或者动摇这个本质。

而钉钉不同。

AI一定会颠覆人们组织办公的方式，如果跟不上，钉钉必然面临淘汰凋亡，此诚危急生死存亡之秋也。不过，挑战伴随着机遇——因为从这条思路出发，整个即时通讯的牌桌也有重新洗牌的可能。

学徒

虽然就从工作时间性价比和人性化程度而言，钉钉都不值得看好，但在面试的过程中，我觉得我有望在钉钉好好的学到一些东西。

一方面，在来到钉钉的那个节点，我自认为有一些审美，但苦手于总感觉还不能很好地把眼睛、舌头上的审美，转化到手上，真正成为创造力和生产力。入职时招我的 +1 是设计出身，他给我的感受是，他有这样的手，以及他不吝让我从他身上好好学到这部分。

其次，更重的原因是，无招确实是一个超级产品经理型CEO。

当时的我，跟过一些leader，也自己做过一些小产品，但始终没有在一个高压高强的环境下，学徒式的学习过。

关键词「学徒」。

做产品和做菜、做手工一样，是一门手艺。有一个综艺叫《一饭封神》，里面许多厨师，流转于许多米其林三星的后厨，积累经验，最终的梦想是独当一面成为米三的主厨，或者开一间自己的餐厅。做产品也一样，在各个大中小司流转，学到本领，最后在哪里栖身独当一面，或者自立门户。

钉钉的面试流程很快，几乎就是一周之内，还包含了一个做大作业的周末。当时的最后一面就是面无招。

可能超级产品经理型的CEO都是这样——后来认识了在理想工作过的朋友，他们告诉我，产品经理去面理想，也要做大作业，也要过范皓宇和李想。

和无招的面试，让我产生一种强烈的直觉，钉钉或许是我的VEA（香港米三，《一饭封神》评委Vicky主厨的餐厅）。它迷人的地方不只在于昂贵、精致、难进，而在于它代表了一种高压、严苛、以及非常私人化的创作现场。那是一个后厨式的工作环境，一个主厨把自己的审美、野心、控制欲和成就欲，全部压缩进一套系统里，然后要求每个进入其中的人，都先接受这套系统的重力。

很多人警告过我他在指导产品工作过程中的暴虐——这点在我正式入职钉钉之后，也有所领略——但在那次面试中，他更多是以一种温和的态度，表达出和对人快速判断的能力和对产品的偏执的理想主义。

面试前半段聊得很顺。我们谈了我在银泰的经历，谈使命感，谈我学习 AI 的背景，也谈到本科心理学。直到聊大作业，我刚解释到「没有完成家族组织」的部分，无招就皱眉打断了我。

钉钉面试前要先完成的那份大作业，题目是「族谱上钉」。要求描述得比较模糊，大意是：把家族成员拉进钉钉，建立一个 6 人以上的族谱组织，让他们在上面真实活动，并基于体验给出产品见解。

坦白说，这件事我没有真正完成。

我家里人比较少，能真正使用钉钉的人更少，无法满足「6 人以上家族组织」的要求。不过我想，做这个作业核心是考察我的产品洞察，敏捷响应，和纳投名状。所以我做了代偿的证明工作：那个周末我去本地图书馆查阅真实的大型族谱，用表格和文档为别人的族谱做编撰和数字化；联系专门做族谱生意的商家，调研商业链条和可能的产品方向；也动员朋友上钉钉，搭了一个出境旅游前事项沟通的朋友组织。

这些工作不算少，我觉得也证明了我的工作能力，并纳上了投名状。

他反复追问：为什么做不成?

父亲家还有人吗？母亲家还有人吗？外公外婆还在吗？真的找不到了吗？真的凑不齐六个能上钉钉的家人吗?

一系列低宜人性的问题，换个环境可能被投诉到政府工会。

在反复追问确认我的显示情况确实凑不齐后，他沉默了大概十几秒，然后开始讲 2014 年他刚做钉钉时的经历。

彼时的钉钉也没有办法证明自己的价值，没有成熟产品和市场共识，只能一遍遍见客户、确认需求、陪笑脸，先承诺、再把东西做出来。大概就是典型的 「fake it until you make it」：在没有路径的时候，先把自己推到必须兑现的位置上。

最后，他给我的判断是：我的「学术底色」太重。

我至今仍不完全认同他的判断，也认为这是一种对人的服从性高于对人的能力的考察。你可以说这是挑选理想主义伙伴，也可以说这是PUA；可以说这是创业精神，也可以说这是权力关系下的过度索取——说到「PUA」，后来他还问了我的婚姻交友情况，以及问我创业有没有做好抛弃这些的准备。

不过，总之，这次面试确实引发了我的反思。

我并不完全同意无招说的，理想主义者就是在看不到希望时，也要坚持往一个方向做。不是所有事情都应该靠意志力硬推，也不是所有不计成本都值得赞美。但我还是被某种信念系统吸引了。我有时候着迷于那种经验事实上成立，但我不能从原理上解释它的东西，比如宗教。它们未必严密，甚至充满问题，但它们能组织人群意志和塑造现实。就像索罗斯的「反身性」原理：市场上的信念足够强烈时，会反过来介入现实，改变参与者的预期、行动和资源分布，最终让现实朝着信念本身靠拢。

「族谱上钉」也是这样，看似荒诞，但却有它实际的用途。换位思考，用最奚落的人际关系来类比，站在王伦的角度，让林冲上山前纳投名状，也是他最低成本判断和提前沟通上山后的工作模式的方式。

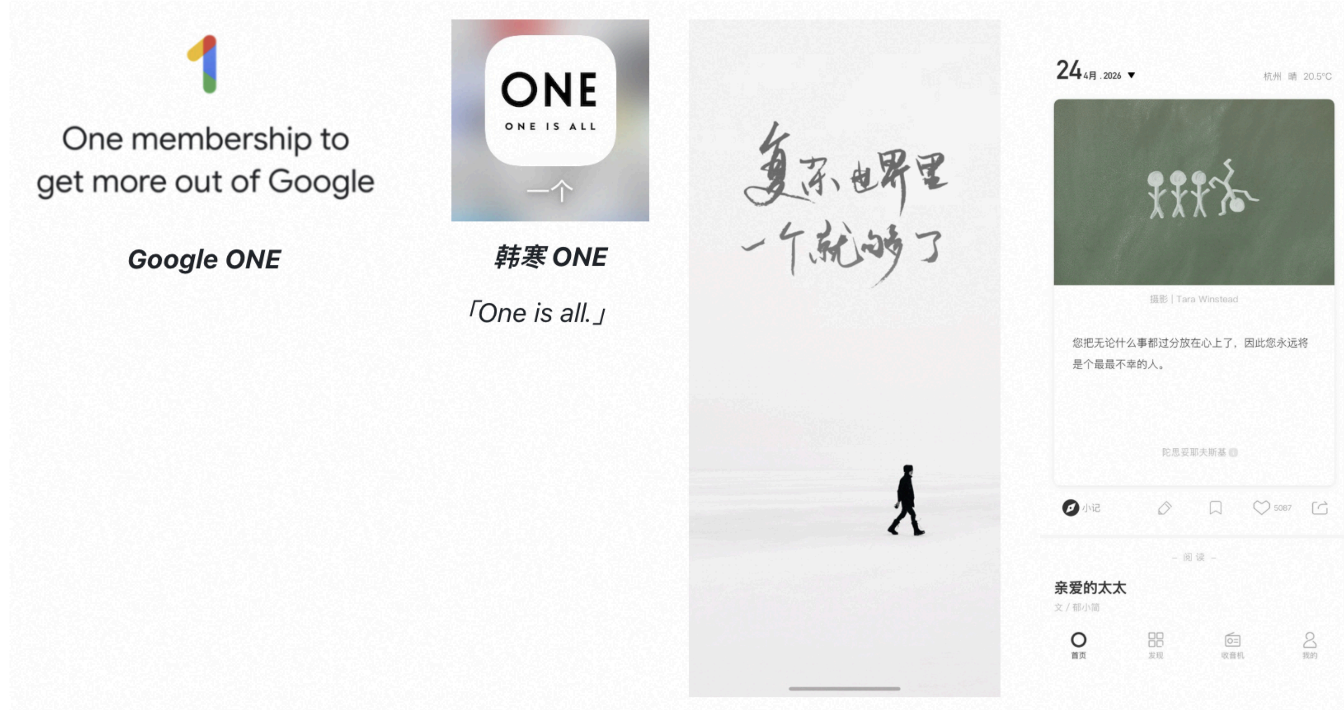
最后他还是通过了我的面试。

总之——在换工作之际，我最大的憧憬可以聚合到三个关键词上：一是context，二是主动服务，三是学徒。他们分别是我认为LLM有效服务成立的条件、我一直想实现的产品形态、和我渴望获得的阶段性成长的环境。

对于我这种一个逐 context （或者说数据）和产品工作而居的游牧民族来说，钉钉似乎都是一片应许的迦南，是撒马尔罕。

ONE 的发心

叫「ONE」的产品有很多，比如同为效率应用方向的 Google ONE，或者韩寒其实也有一个 app 叫 ONE，slogan 是「复杂世界里，一个就够了」。



ONE 的愿景发心，和所有叫「ONE」的产品一样：「all in ONE」，所有工作都在一页内处理。

更具象地拆，可以分成几层。

第一层，是用户发心。钉钉里的工作信息太散，群聊、待办、日程、会议、文档、审批各管一段，用户每天靠自己的记忆和注意力，把它们勉强串起来。ONE 想解决的是这个问题：让重要的事自己浮上来，让用户少一点遗漏，少一点翻找。

第二层，是产品发心。钉钉需要一个 AI 时代的新入口。它不能只把 AI 做成聊天框，也不能只把 AI 藏在各个老功能里。ONE 要把钉钉已有的消息、日程、待办、会议、文档重新摆一次，让外界看见：钉钉不是旧办公软件，它还能重新组织工作。

第三层，是组织发心。无招回归后，钉钉需要一场新的战役来聚人心、提士气、改形象。ONE 承担了这个角色。它既是产品，也是旗帜。旗帜能聚拢人，也容易把太多东西都挂上去。

第四层，是商业发心。AI 要消耗 token，模型能力也需要场景消化。这个阶段有点像发动机刚被发明出来，全世界都在找哪里可以装发动机。钉钉有大量工作场景，如果 AI 真能帮用户提升效率，当然是用户、产品、集团三方共赢。

这些发心若再压缩，可以归成几条：解决深度用户工作信息爆炸的问题，提升工作效率，更好地服务 KP 人群，销售 token 资源。哪怕最后一条是从集团使命出发，也有它自己的顺理成章。刚有了发动机，世界当然会寻找发动机的用武之地；如果它确实让车跑得更快，让船行得更远，那就是完美共赢。

所以 ONE 一开始就不是一个发心单纯的产品。它既想替用户减负，也想替钉钉换代；既想证明 AI 能进入工作流，也想证明钉钉还能站在下一代办公入口上。这里面有理想，也有指标；有用户价值，也有组织意志。

后来很多问题，都可以从这里追溯：ONE 究竟先服务用户减负，还是先服务发布会表达？先做所有人的入口，还是先做少数深度用户的真需求？先让事被看见，还是先让事被做完？

这些问题，在「发心」章里只是种子，到了后面的「定位」「设计」「用户」「敏捷」，它们会一层层萌芽、抽条、长叶、开花，结出完整的产品果实。

定位第二

如「发心」章所述，ONE 最吸引我的地方，在于钉钉丰饶的数据土壤，以及「主动服务」这个产品理念。

但发心只是起念，是马丁路德金的「I have a dream」，发散，抒情。一个产品真正进入开发，要回答的不是「想做什么」，甚至不是「要做什么」，而是「先做什么」。

所以，发心之后必须有定位。

如果把一个产品从虚到实的过程，比作拾柴点火，发心是选柴，定位就是捆柴的绳。柴选得好，只说明有得烧；捆不好，设计进来就是到处点火，这一丛，那一丛，看着热闹，风一吹就散。用户用不上、记不住，团队也徒耗力气和信心。

ONE 后来的许多问题，表面上出在设计、用户反馈和开发节奏里，根却埋在更早的定位问题：到底为谁服务，在哪个场景里用，解决什么价值，又凭什么由钉钉来做。

本章先不急着讲具体功能。我们先回到产品落地的第一步，看烧起 ONE 的这把柴火，最初究竟是怎样被捆起来的。

什么是产品定位

产品的定位其实可以拆为一系列关键的产品策略，一系列彼此解耦的主观选择和假设。

有的假设是客观题，有的则是主观题。主客观之间的底层差异在于，客观题是可以通过物理规律验证的，而主观题则符合索罗斯的反身性定律——好比股票市场，多数人的信念会反向影响现实。客观题到了一定阶段可以证伪，比如行业垂类小模型；而主观题无关真伪，纯粹是品味和选择，比如：

- 苹果选择闭源，安卓选择开源，
- ChatGPT、豆包选择大 DAU，Anthropic、Manus 选择高净值塔尖用户，

闭源还是开源，DAU还是塔尖用户——都没问题，都是选择，想清楚且走到底，都有可能成为开宗立派的现实。反过来说，如果一直在多种底层策略的判断上逡巡不定——那么从用户侧看来也会不伦不类，产品迭代起来鸡犬不宁。另外从资源效能和团队信心角度出发，也是极大的浪费。

更具体地，要说清产品定位，需要明确回答四个问题：

1. **用户定位——"为谁"**
    1. 目标用户是谁？不是谁？
    2. 用户的关键特征是什么（规模、行业、角色、习惯）？
2. **场景定位——"在哪用"**
    1. 用户在什么情境下想到这个产品？
    2. 是高频日常工具，还是特定场景的解决方案？
3. **价值定位——"解决什么问题"**
    1. 产品提供的核心价值是什么？
    2. 用户原本愿意花费多少钱/时间/其他资源，来达成这种价值？

        4. 竞争定位——"为什么是我们"

    3. 和替代方案相比，差异化在哪？
    4. 这个差异化是否可持续？

当这些问题能够一句话说清时，产品就有了清晰的定位。

ONE 的产品定位

无招在 2025 年 4 月 1 日回归钉钉， 8 月 25 日开了第一次产品发布会，版本8.0，主题「蕨」。

按照第一次发布会的口径，ONE 是「钉钉面向 AI 时代打造的、以 Agent 驱动的工作信息流」。更详细的阐释是：ONE 是一次试图把群聊、待办、会议、文档、表格等分散工作信息重新组织起来的尝试——让 AI 不只是被动回答，而是主动替用户发现、整理和推进工作。

我们当时的 slogan 叫，「让『人找事』变成『事找人』」。

按照上述四个定位问题，来整理 ONE 的定位，大致如下：

定位问题	钉钉 ONE
为谁	高信息密度、高协作负担、对消息/日程/任务闭环负责的钉钉用户
在哪用	呈现钉钉内的工作信息流，尤其是消息、日程、待办、听记之间
解决什么问题	消息太多、事项太散、重点不清、容易漏回、任务难闭环
为什么是我们	钉钉掌握组织关系、消息、日程、待办、会议、企业协作链路，比通用 AI 有更贴近真实工作现场的丰富 context

然而，分析是清晰的、指日可待的，现实却往往是泥泞的、出乎意料的。就像项飙在《跨越边界的社区》中写道：「不管是一个人还是一个民族，发展主要来自不断的探索，而不是事先一揽子的设计。」

2025 年，正值 LLM 从 demo 走向真实工作流动荡而迅猛的上升期，能力令人兴奋，但稳定性、评测、成本、延迟、操作权限、用户隐私等等都还是一团草莽，没有成熟机制可以参考模仿。

同时，钉钉作为企业协作产品，本来就天然同时服务组织中多组相对角色的目标、管理者和员工、发信人和收信人，如何在相对角色中设立公平范式，还在探索期；而 ONE 又承担了对外宣发、对内集团指标、对客服务、商业化探索、及产品形象创新换代的多重目标，无法像一个独立创业产品那样，看住一个窄切口慢工打磨。

ONE 看似是一个很清楚的产品：一个 AI 工作信息流，一个专属工作秘书，一个让事找人的入口。但越往下做越会发现，它真正的难点不是功能实现，而在于要在一块小小的移动端屏幕上，同时回应管理者对全局掌控感的渴望、员工对信息过载的抗拒、以及AI对「什么值得我关注」这一判断本身所携带的权力属性。一个「让事找人」的入口，一旦运行起来，就不得不替组织回答：谁的事算「事」？在谁的优先级序列里插队？AI替谁节省时间，又替谁增加可见度？ 更深处，LLM 的不确定性让这些张力被急剧放大。当模型幻觉可能出现在审批意见、周报总结、跨部门同步中时，「秘书」的可靠性不再是技术指标，而变成了组织信任问题。成本约束要求我们对每一次「主动推送」精打细算，可重要信息恰恰可能因为一次节省资源的静默而被错过。隐私边界在多角色环境中没有明确标线：为员工提供更好的上下文，往往意味着让更多数据对模型可见；而向管理者透明化过程，又可能滑向微观监控。 总之，ONE 不是先定义好再执行的产品，而是在多方目标拉扯、技术草莽与现实泥泞中，一边建立判断力一边长出来的工作界面。

所设想的典型工作场景

在场景定位上，ONE 的核心假设非常纯粹：**用 AI 帮用户整理工作并重新排序**。团队并没有把功能零散地扔给用户，而是紧扣着职场人一整天最高频、最容易产生信息焦虑的三个特定时间节点，拎出了三段式的「时间线场景」：

- **场景一：高压之后的「补课与跟进」(信息堆积场景)**



- 这是职场人最常见的痛点。当你开了一个两小时的长会，或者半天没看手机，再次打开钉钉时，面对的是几十个群里炸开的海量未读消息。此时用户的核心诉求是「快速赶上进度」。ONE 的场景定位就是在

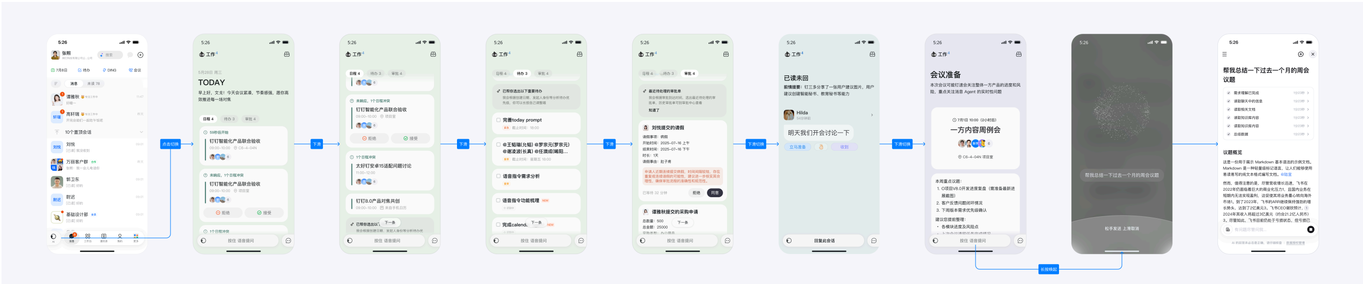
这个瞬间充当智能过滤器，不让用户逐条去翻聊天记录，而是把堆积的信息自动排好优先级，像呈送摘要一样让用户一目了然。

- **场景二：下班前的「查漏补缺」（收尾复盘场景）**



- 在一天工作即将结束、准备下班的节骨眼上，用户需要有一种「今天的事情都交代好了」的确定性。ONE 在这里的定位是安全网。它会启动「已读不回」的智能提醒，防止你遗漏重要客户或老板的信息；同时，它会把今天所有群聊里的金句、核心技术结论、重大业务决策进行汇总总结，让用户气定神闲地完成今日闭环。

- **场景三：晨间的「排兵布阵」与「发现」（开启新一天场景）**





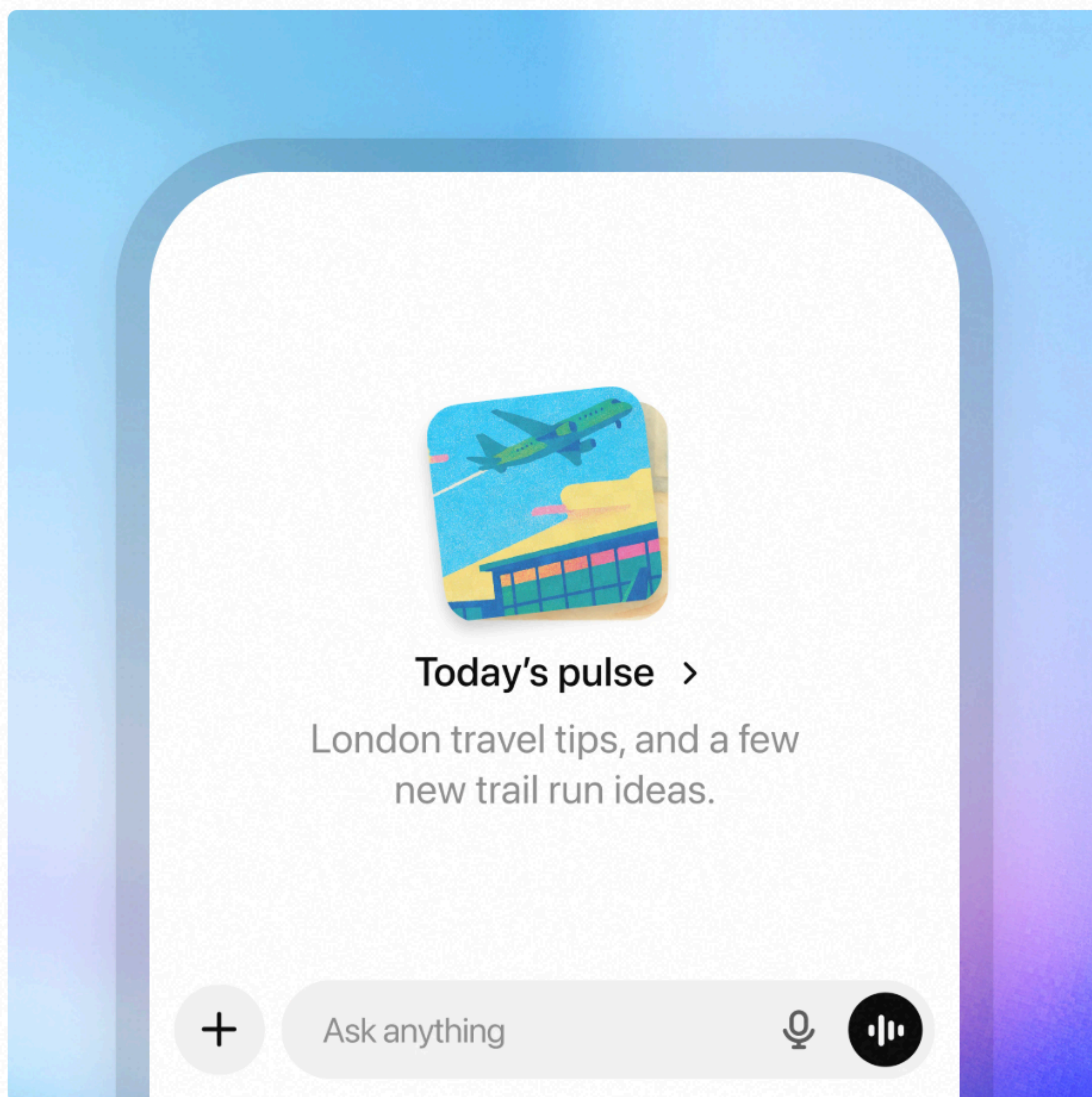
- 这是用户每天早上开启工作前的准备阶段。ONE 会在清晨为用户梳理好这一天的整体工作待办、准备好即将开始的日程，做好优先级的智能排班。在这个场景的尾端，产品还长出了一个具有前瞻性的「发现」

功能——AI 会根据用户的日常工作上下文，主动推送你可能需要阅读的学术 Paper、竞品动态或行业新闻简讯，作为开启一天的认知燃料。

这三个场景在底层是高度自洽的，它们本质上都是在用户最需要组织生产力的时候，用 AI 的「事找人」去对抗无序的信息重力。

2025 年 9 月 25 日，OpenAI 推出了同样主打「主动服务」的 ChatGPT Pulse：

每天更新，从不缺席

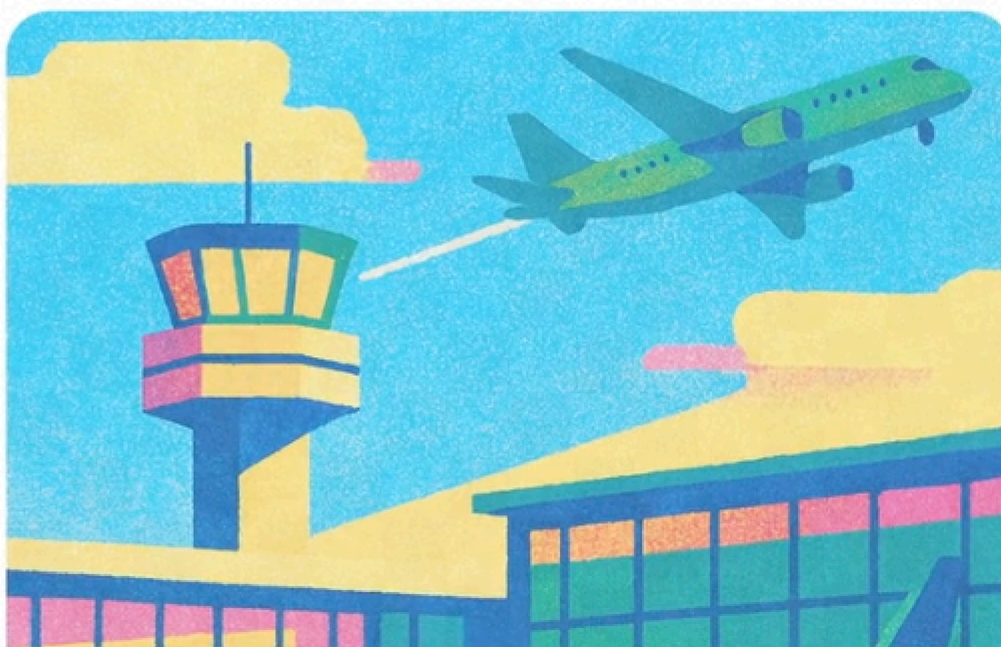


ChatGPT 现在可以代替您进行错时研究。每天晚上，它都会综合您的记忆、聊天记录和直接反馈提炼出与您最为相关的信息，然后在第二天为您提供重点突出的个性化更新。具体内容可能包括您经常谈及的话题跟进，简单又健康的晚餐建议，或者铁人三项训练等长期目标的后续步骤。

您还可以连接 Gmail 和 Google Calendar 来提供补充背景信息，这样它就能提出更具相关性的建议。连接日历之后，ChatGPT 可能会起草一份会议日程样本、提醒您购买生日礼物，或者为即将开启的旅程推荐备选餐厅。这些整合功能默认处于关闭状态，您随时可以通过设置开启或关闭。

September 25

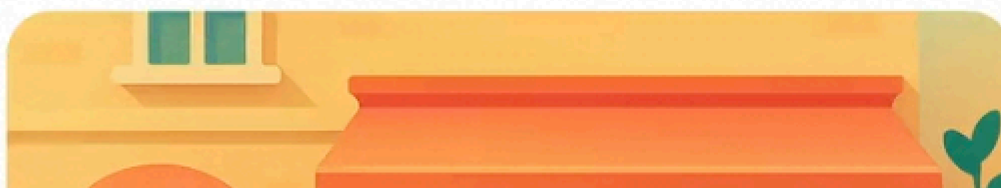
Hey Aleks, your big trip is around the corner. Here's a handy guide for Heathrow and a few dinner ideas.

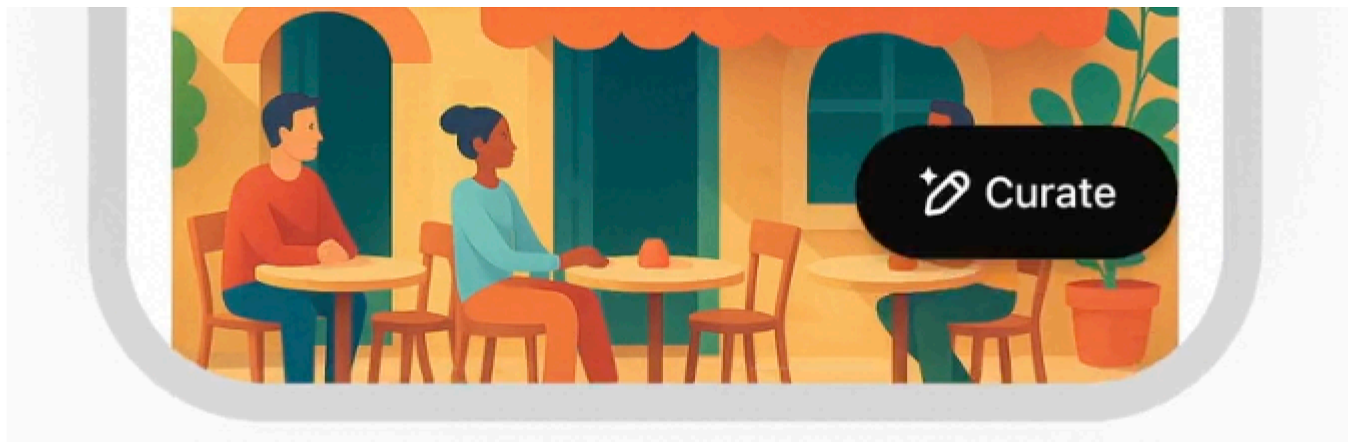


A Guide to Europe's Busiest Airport

Check Your Terminal: Heathrow has four terminals, and they're far apart.







Pulse 宣称打破了传统的「人问 AI 答」的被动范式，通过理解用户在 Google 全家桶上的日程、邮件等上下文，利用 AI 在用户熟睡的深夜进行异步研究，在清晨为用户呈上一套由 5 到 10 张视觉卡片组成的「Today's Pulse」早报。

Pulse 的出现，印证了 ONE 团队对于「清晨沙盘+主动发现」的场景直觉，是站在全球 AI 生产力工具前沿的。

两个影响用户定位的关键决策

有两个影响后续一系列设计开发的问题，在产品定位这个阶段就埋下了祸根。这两个问题都是用户定位的矛盾，分别是：

老板 v.s. 员工

在钉钉上，老板和员工是截然不同的画像。

1. 老板，或者说管理者，是工作的发起者，需要对接更多的员工，督办、催办员工完成任务；他们有大量的日程，他们处理员工发起的审批，给员工布置待办。

相应的，员工是承接方和任务的完成方。他们一方面，打卡考勤，接受老板的任务和监督，向老板提出审批请求，完成老板的待办，另一方面，他们需要真正去填报表、写文档、联系客户。

举一个更有画面感的例子：员工给老板发语音消息的比例，是老板给员工发语音消息的零头。

面对这两种不同的用户画像，先服务好谁的诉求，必须有所取舍。

ONE 的服务对象，究竟是老板还是普通员工，是管理者还是被管理者，其实都有各自的理由。

先说服务员工。

前文已经说了，ONE 的目标是服务大 DAU 且高频的用户操作。如果只服务管理者这样的「塔尖」用户，就达不成给发现带量，或者卖 tokens 的目的。

另外，企业购买一个 SaaS 产品的本质，是为了更好地组织和管理自己的员工，确保每个员工可以上下一心，更高效地完成任务，达成组织目标。仅服务管理层，价值太薄。

那为什么又要服务老板呢？

做产品一方面看用户要什么，一方面看我们有什么。

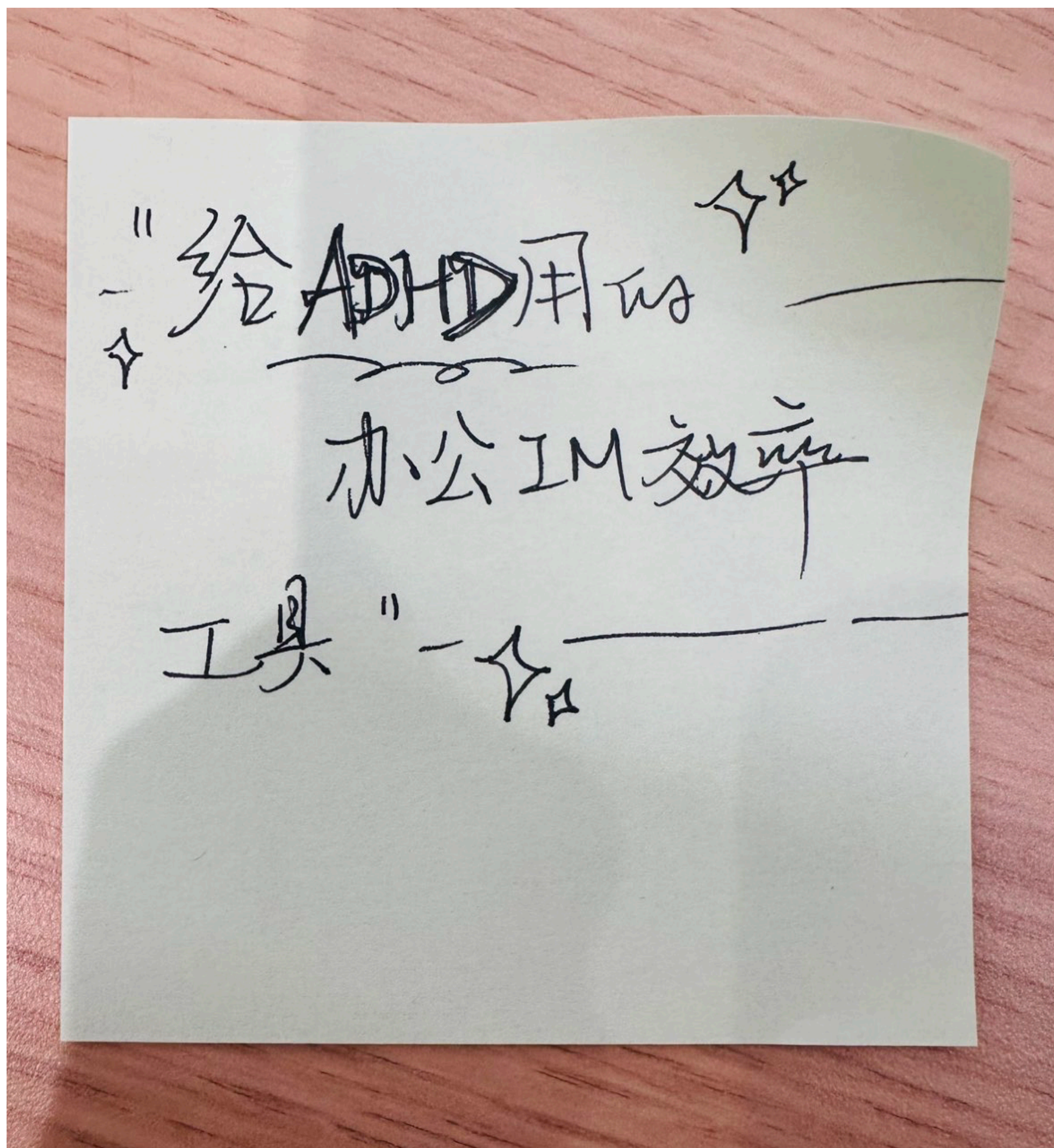
2025 年 7 月 9 日，飞书开了发布会，重推 Aily（飞书的 AI 助理）。会上谢欣借讽「某 app 上最多的数据就是各种打卡」，力陈飞书才真正重视用户在办公场景中的信息流转和知识沉淀，更适合 AI 时代。

确实，钉钉早期推出已读未读，重搞 DING、考勤打卡、审批流等品，就是依靠服务企业管理层的监控和管理需求起家。钉钉的文档表格知识库是标准品，但是在飞书出现后，历史上长期有一段时间，这些品的基建和交互体验都是弱于竞对的。

同时反向来说，由于如上的产品特性，制造业、传统企业、以及更多没有很强知识管理诉求的企业，会选择钉钉；服务好这部分企业，又让钉钉不断朝着这些企业需要的方向进化。由此形成了飞轮。

钉钉上的信息枢纽，绝大多数还是管理层，他们有最多的 IM消息，日程，待办，审批。

ONE 目标对标高级机要秘书的职能：解决信息过载，让组织中的重要事项不再遗漏，通过智能排列工作信息的优先级，来节省人的精力、注意力。在封闭开发过程中我还开过玩笑，以后宣传口径就用「ONE 彻底治好了我的工作 ADHD」。



在钉钉，究竟谁有这么多工作需要批阅处理?

那也就是老板们了。

到这里，产品定位似乎被逼进了一个角落。

从产品目标而言，似乎应该服务钉钉的所有用户；但从交付的维度而言，又要从老板们服务起。那么，我们也就只有一个指望，就是通过老板 KOL 们的带动，让越来越多的员工用户，可以并愿意在钉钉处理所有的工作内容，并且把越来越多的信息沉淀到钉钉上。放眼同时期钉钉开始在做的录音卡硬件 A1，也都是为了协同完成让信息上钉更加方便容易的目标。

在初始定位的构思中，用户究竟是普通员工还是老板的问题，始终没有闭环。

但是事情总不能无休止地讨论下去，我们就这样，带着一盒薛定谔的用户出发了。

发信人 v.s. 收信人

钉钉和微信最大的区别是什么?

微信的立场是「收信人」。为了保护收信人的不被打扰，微信可以克制地不做已读未读，不做强触达。而钉钉的基因，从诞生的第一天起，就是永远站在「发信人」立场、为「发信人」所驱策的。

在进行 ONE 的产品讨论时，这个点也被反复提及，具体会在下面「设计」第三章，讲「已读」的时候细讲。

如果说有三种产品经理类型：布道者型/服务商型/朋友型，他们分别可以对应乔布斯，贝尼奥夫，和雷军，那么无招是典型的第一种，一个极具煽动力的典型布道者。当他在会议室里讲到十年前如何顶着压力做 DING、做已读未读时，你几乎能从那些话语碎片里，看到老钉钉一以贯之的权力美学——站在发信人一侧，替组织争取绝对的确定性，用强触达把事情硬生生往前推。

在企业协作的生态位里，发信人的诉求，往往与管理者的控制欲和对确定性的焦虑高度咬合。管理者购买钉钉，核心痛点是消解「我说的话，对方到底看见没有；我交代的事，到底有没有往前走」的确定性焦虑。

这套身体记忆是如此成功，以至于它不可避免地流淌进 ONE 的血液里。

为什么卡片里的消息一定要算已读? 为什么系统要主动把事情推到用户面前? 为什么 AI 忍不住要替组织去催促个体? 很多看似属于交互细节的争论，答案都可以回到这个原点。

然而，ONE 从概念阶段开始，就试图去承接一个宏大的、甚至带有平权色彩的 AI 愿景：通过「事找人」来帮员工减负。这就让 ONE 必须去面对组织中硬币的另一面——**收信人，同时也是普通员工，或者说沉默的大多数。**

这就带来了用户定位上的第一个内在张力：一个产品如果骨子里流淌着「发信人立场」的强控制基因，它的功能设计和算法逻辑就会不自觉地倾向于催办、督办和绝对触达；但它在对外叙事和价值承载上，却又试图去扮演一个体贴「收信人」、帮收信人过滤噪音的超级秘书。

这种基因与愿景的底层拉扯，让产品在定义「我到底在为谁说话」时，从一开始就埋下了失焦的隐患。

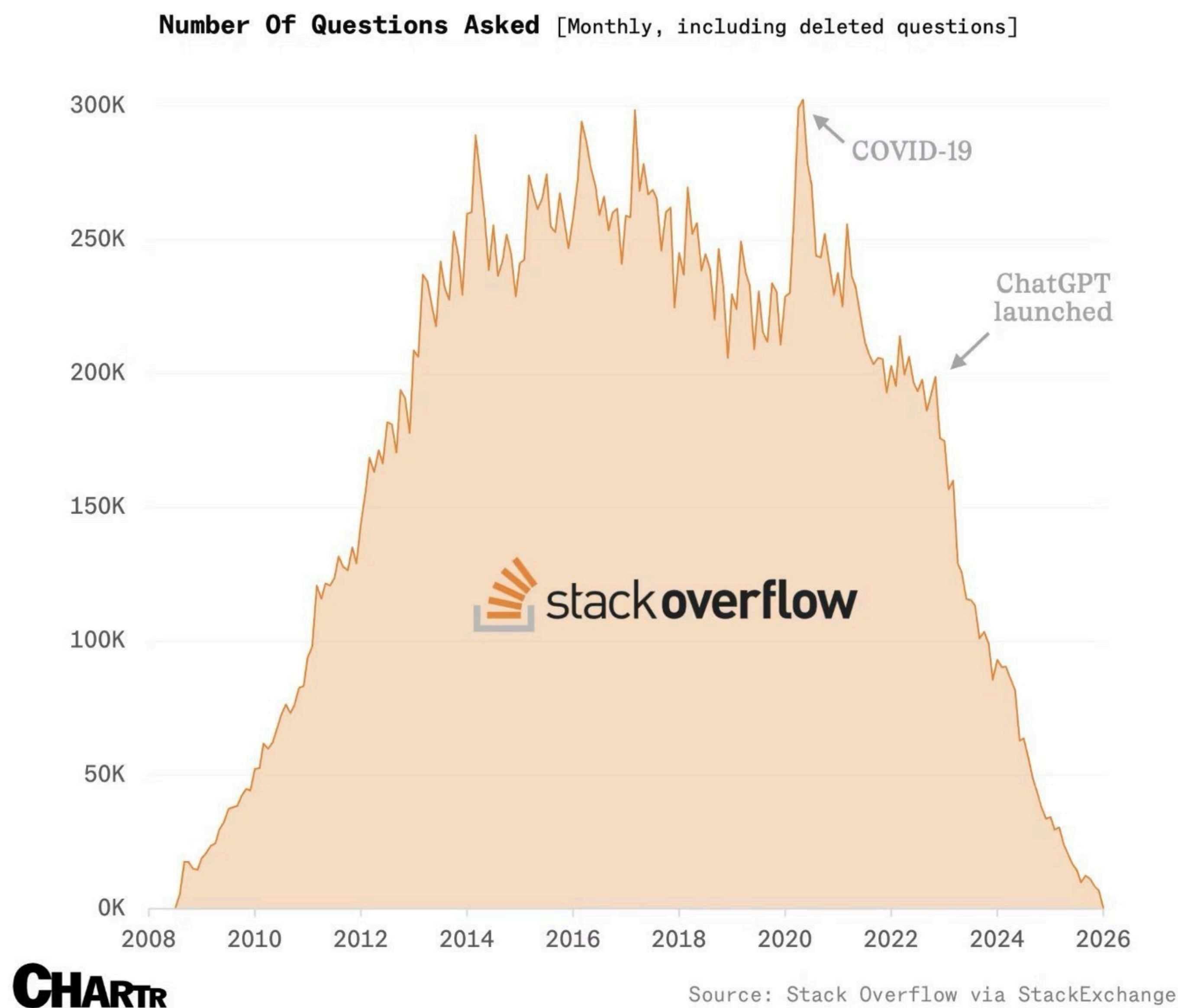
当 AI 成为发信人的超级代理人，自动整理出待办、未读并呈现在移动端时，收信人（普通员工）感受到的并不是「智能的平权」，而是「被凝视的加剧」。AI 削弱了员工在工作流中用于喘息的「心理缓冲带」。普通员工变成了系统内的沉默大多数，他们的反抗通常极其隐蔽——比如在社媒上写劝退帖，或者在实际使用中寻找各种手段去糊弄和规避这个随时会跳出来的 AI 界面。

竞争定位——"为什么是我们"

I. 为什么是钉钉

ONE 的竞争定位，不能只看「谁也在做 AI 助手」。如果只是比模型能力、对话体验、生成质量，钉钉并不一定天然占优。真正的问题是：在 AI 时代，钉钉凭什么继续成为工作入口？

这里可以看 Stack Overflow 的曲线。ChatGPT 出现之后，Stack Overflow 的提问量急剧下滑。这张图对所有工具型产品都是一个提醒：当 AI 能直接给答案时，用户不一定还会回到原来的社区、流程和入口。过去积累起来的内容、关系和使用习惯，并不会天然构成护城河。它们可能反过来成为被 AI 抽象、吸收和绕过的对象。



对钉钉来说，挑战也是类似的。AI 不是给旧产品加一个聊天框，而是在重新分配入口。用户未来要完成工作，未必一定先打开某个应用、进入某个页面、点击某个功能。他可能直接对 AI 说：帮我总结会议，通知相关人，生成待办，更新项目进度，提醒我明天跟进。这个时候，谁掌握组织上下文、谁能调用真实权限、谁能完成闭环，谁才有资格成为入口。

钉钉的优势正在这里。它有组织关系，有消息链路，有日程、文档、会议、审批、待办，有企业通讯录和真实工作流。它不只是一个信息容器，而是企业协作的操作现场。理论上，ONE 可以利用这些上下文，把 AI 从「回答问题」推进到「理解工作」和「推动协作」。

钉钉来做这件事，有一个非常强的理由：它有真实工作现场。

一个普通 AI 助手很难知道组织里谁和谁有关，哪条消息重要，哪个会议有后续，哪个任务真的会影响项目。钉钉知道。它有通讯录，有群聊，有日程，有文档，有审批，有会议，有待办，有企业里的关系和流程。

这些是钉钉的优势。

但这些也是钉钉的重量。多组织、权限、历史链路、端侧差异、客户定制、旧交互习惯，都会变成 AI 产品的成本。越想进入真实工作流，越要面对这些东西。

我们也和一些大客户聊过，想看 ONE 的能力能不能覆盖他们已有的任务工作台和定制面板。结果很快发现，两者并不完全匹配。大客户要的是贴着自己业务长出来的东西：自己的流程，自己的字段，自己的任务面板，自己的管理动作。ONE 要做的则是一个面向所有钉钉用户的轻入口，不是专属钉式的定开功能。

这带来一个很现实的问题：钉钉是 SaaS，ONE 却带着很强的 To C 入口想象。

做得太广，就只能抓通用痛点；通用痛点又容易不够痛。做得太深，就会变成客户定制，失去全局入口的意义。ONE 卡在中间：它想成为所有人的入口，又很难替任何一类人把深问题做透。

II. 为什么是 ONE ——或者说，为什么不得不用 AI

讨论 ONE 的竞争定位，不能只问「为什么是钉钉」，还要问「为什么是 ONE」。

钉钉有组织关系，有消息、日程、会议、待办、文档、审批，这些确实是它做 AI 工作入口的底气。但 ONE 并不是从零开始争夺用户。它首先要面对的竞争对手，恰恰是老钉钉。

用户已经有自己的工作路径：消息去消息页看，日程去日程页查，审批去审批中心批，待办去待办列表处理。这个路径未必优雅，却足够熟悉。对很多用户来说，旧系统的最大优点不是先进，而是可预期。入口在哪里、红点怎么看、消息点进去会发生什么、审批点同意后是什么结果，用户心里有数。

所以**如果用户在老钉钉里依靠「肉眼扫视、手动标记、特意置顶」就能应付日常工作，他们就绝不会付出注意力和信任成本去迁移到一个新卡片流上。** ONE 必须解决一个「不依靠 ONE，用户在老钉钉里就算用断手也无法解决」的场景问题，才能击穿这里的迁移成本。

这里其实有三层迁移成本。

第一层是入口迁移。用户原本打开钉钉，第一反应是看消息页。ONE 如果要成为新首页，就要让用户相信：从这里开始，比从消息页开始更省事。

第二层是心智迁移。老钉钉是按工具分工：消息、日程、审批、待办，各自有边界。ONE 是按事情重新组织：哪件事重要，哪件事该看，哪件事该跟。这个心智更先进，但也更不稳定，因为它要求用户相信系统的判断。

第三层是责任迁移。老路径里，用户自己决定什么时候点开消息、什么时候进入会话、什么时候处理审批；ONE 主动把事情端到眼前，等于提前改变了用户和工作的距离。系统越主动，用户越会问：你凭什么现在推给我？我看见之后，会不会马上承担责任？

这个**「不依靠 AI 就无法解决」**的场景到底是什么？

- **老钉钉的局限（旧范式的死穴）：** 传统的 IM（消息流）是**一维的时间线线性结构**。当一个用户面临 50 个活跃群聊、每天上万条信息时，无论他的肉眼有多快、操作多熟练，他也只能「一条一条地看」。在旧系统里，信息是「平铺直叙」的，前线 BD 聊出的一个重大客户痛点，和群里的一句「收到」、「哈哈」，在 IM 界面上占据的空间和红点权重是完全一样的。

- **非 AI 无法解决的场景：高密度的「非线性逻辑缝合」。** 比如你在 7 月 3 日遇到顾家家居的痛点：前线销售和供应链的协同，消息散落在「顾家生产群」、「顾家交付群」、「顾家销售群」等 5 多个不同的群聊里。原本老钉钉做得再好，员工也必须切换 5 个窗口，用大脑进行人肉记忆、联想、拼凑，才能发现「哦，原来生产群里提到的延期，是因为销售群里昨天客户改了需求」。**这种跨群聊、跨时间、跨文档的「上下文网状逻辑缝合」，在老钉钉里，用户就算用断手，也只能靠大脑肉搏。** 只有通过基于 ONE 的 LLM推理，才能做到「把散落在不同时空碎片里的冰鞋线索，自动编织成一条线送给用户」。

这些才是 ONE 相比老钉钉可能成立的地方。

如果 ONE 只是把老钉钉里已有的东西换一种样式摆出来，用户很难迁移。因为原路径虽然旧，却已经够用；新入口若只是更好看、更聪明一点、更主动一点，不足以抵消迁移成本。尤其对企业软件来说，「够用」本身就是很强的惯性。

这也是 ONE 的竞争定位中最容易被低估的一点：它不是在和一个空白市场竞争，而是在劝用户离开一个他们已经习惯、已经会用、虽然抱怨但仍然依赖的旧系统。

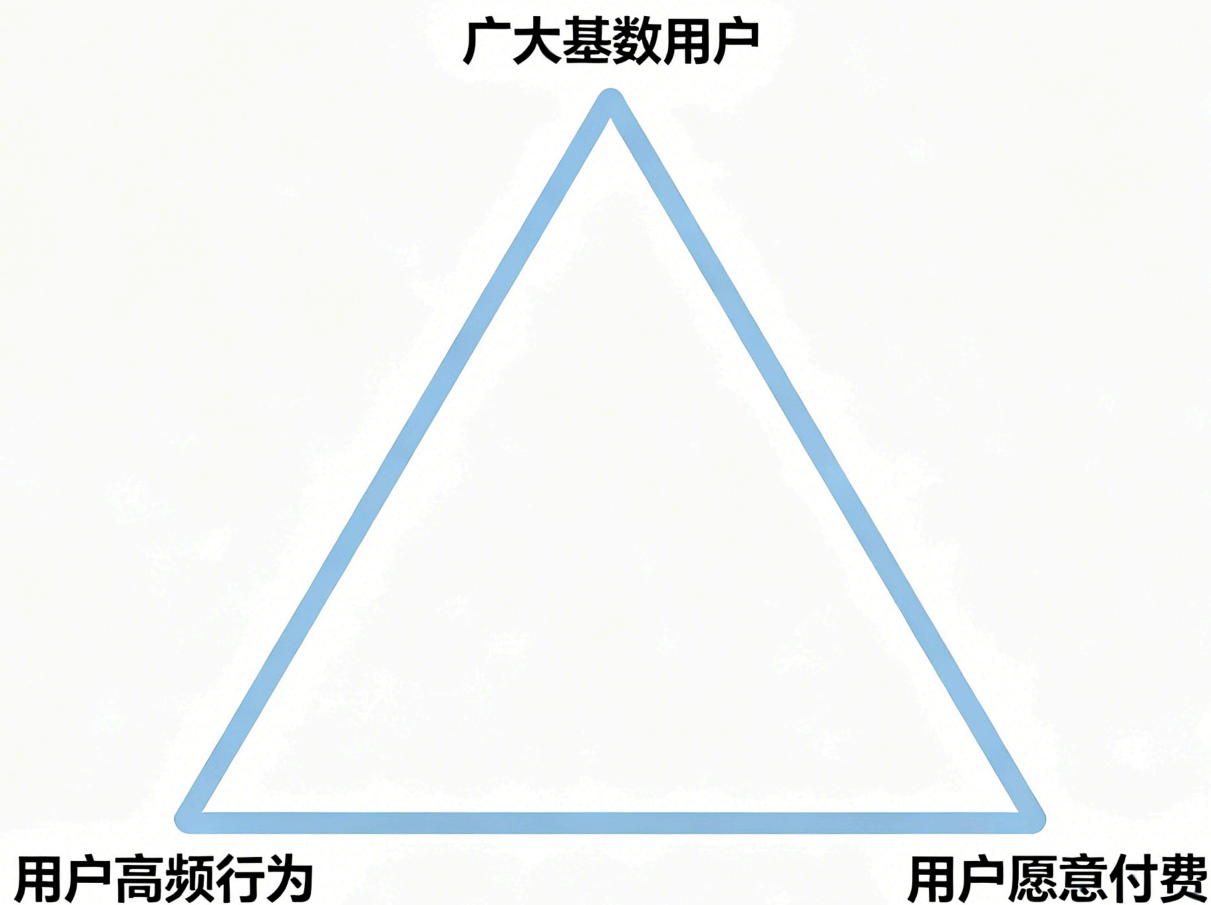
所以，「为什么是 ONE」的答案不能停在「因为钉钉有数据」。数据只是资格，不是胜势。ONE 必须证明：它不只是钉钉旧能力的一次重新摆盘，而是真的解决了老钉钉解决不了、或者解决得太费力的问题。

换句话说，ONE 的竞争对手有三类：除了外部的两类——飞书、企业微信这样的办公 SaaS 软件，和各种 AI Agent；还有内部的老钉钉：这是更深的一层，要和用户已经形成的工作习惯竞争。外部竞争决定钉钉为什么要做 AI，内部竞争决定用户为什么要从 ONE 开始工作。后一件事，比前一件事更难。

动态变化的定位与阶段目标

在整个开发周期中，定位和对应目标多次阶段性地标调整过，每次调整，都对应着一些不得不解决的现实问题。

最早摆在团队面前的，不是某一种具体形态，而是一个很难同时满足的三角：广大基数用户、用户高频行为、用户愿意付费。



- **不可能N角**

ONE 期望多个目标之间同时找解法：

- 用户只需要用 ONE就可以完成在钉钉上的所有操作
- 用户每天高频使用
- 辐射大多数用户
- 服务好管理者角色
- 成本控制
- 引导用户去看资讯视频
- AI 召回率、准确率。

最核心的处于量子纠缠态的目标三角是：

目标	对应诉求	问题
广大基数用户	要够轻、够泛、够高频，最好每天打开	痛点容易浅，用户不一定愿意迁移
高频工作行为	要嵌入消息、日程、待办、会议这些每日动作	容易变成老钉钉能力的重新摆放
用户愿意付费	要足够深、足够具体、最好能解决业务问题	越深越像定制，越难做成所有人的入口

这三件事很难同时成：

如果要大基数，就要抓广谱痛点；广谱痛点通常不够痛。

如果要高频，就要进入消息、日程、待办、会议；这些老钉钉本来就能做，迁移理由不够强。

如果要付费，就要解决深问题；深问题往往长在行业、业务、组织习惯里，很难靠一个通用入口解决。

所以 ONE 早期真正的定位难题，是在「所有人都会用」「每天都会用」「有人愿意付费」之间找一个交点。这个交点本来就很窄。

如果要服务广大基数用户，产品就要轻，要泛，要让足够多人每天都能打开；如果要抓高频行为，就绕不开消息、日程、待办、会议这些老钉钉本来就有的东西；如果要让用户付费，又必须解决更深、更具体、更贴近业务的问题。三者都重要，却很难靠同一个入口同时满足。

这也是 ONE 早期定位最难的地方。它既不想只做一个小工具，也不想变成客户定制；既想覆盖所有钉钉用户，又想做出足够深的 AI 价值；既要让用户高频使用，又要证明这个入口不只是老钉钉换了一种摆法。

- **用「工作 + 发现」试图同时解三角**

后来「工作」和「发现」的组合，可以看成对这个三角的一次求解。

「工作」负责高频和大基数。消息、日程、待办、会议每天都发生，只要能把它们重新组织起来，ONE 就有机会成为用户每天打开的入口。「发现」则承担另一层想象：学习内容、知识流、paper、视频化内容，理论上更接近收费场景。于是 ONE 既包揽钉钉全局信息，又试图用工作流给发现带量。

这个组合看上去完整。工作带 DAU，发现带收费，ONE 做统一入口。但它也把两种意图放进了同一个页面：一种是处理工作，一种是消费内容。用户打开 ONE 时，最朴素的意图往往是看今天有什么事要处理，而不是突然进入一个学习流。于是「发现」在产品逻辑里负责商业想象，在用户体感里却容易变成不合时宜的打扰。

- 「工作」模块负责广大基数和高频行为；
- 「发现」模块负责收费想象；
- ONE 作为统一入口，把钉钉全局信息都收进来，试图既有 DAU，又有商业化，又有 AI 新入口的故事。

这个思路内部是自洽的。

工作信息天然高频，消息、日程、待办、会议每天都发生；发现内容更接近学习、成长、知识付费，理论上有收费空间。两者放在一个入口里，似乎就能让高频带动低频，让免费带动付费，让工作流带动学习流。

但实际问题是，用户打开 ONE 的首要意图并不一定接受这种拼接。

他来处理工作，不一定想学习；他来清消息，不一定想看 paper；他来找今天该做什么，不一定想进入一个内容流。于是「发现」不是自然长出来的付费场景，而更像被工作流带量的商业目标。这个判断要写得克制，但可以明确。

这一段的核心不是「发现不好」，而是：

发现把商业发心提前塞进了工作入口。

用户看着某某台的电视剧，中间突然插进来一条某某乳业的广告——你可以说它的发心是给中国人推荐好牛奶，但用户能不知道这他妈是广告吗?

发布会之后，ONE 又被寄予了更明确的指标。产品不再只是讲一个 AI 工作入口的故事，还要回答有没有人用、能不能提升效率、能不能贡献 AI 组织数。指标进来以后，定位自然会向更容易被计量的部分靠拢：曝光、打开、点击、AI 调用、组织覆盖。

只是，真正深的 AI 价值往往更慢，也无法通过这套指标衡量。用户希望 AI 帮他建待办、做搜索调研、跟进事项、跨系统执行、按自己的规则处理一类问题；但这些能力牵涉权限、稳定性、成本和长流程执行，短期内不容易被做成一个轻入口。于是 ONE 在现实推进中，更容易停留在让事情被看见、被整理、被触发，而不是让事情被完成。

到了 2025 年 9、10 月份，随着发布会落幕，由于外界刺激与对赌指标的非连续性，原本的 ONE 路线开始边缘化，并出现了不小的人员流失。

到了 11 月中旬，ONE 原本「提纯信息、对抗过载」的严肃生产力中枢定位被完全解构，让位给另一套更大的说法：AgentOS ，底部层层堆叠的一排**对于普通打工人而言完全用不到的 Agent 图标**，不仅在内网引发了剧烈的反弹，更遭遇了不少客户的直接质疑与投诉。



但它在概念层面上，完成了决策者个人对于「AI 操作系统」的**宏大美学投射**。这也让一线产研更加意识到，钉钉已很难长出真正刷新 AI 时代、尊重个体自主性的 AI 工作产品。

新的关键词已经变成 LUI + GUI、Agent OS、统一 AI 入口、多 Agent 运行态。消息、日程、考勤等高频功能，都被重新理解为可以运行的 Agent；未来的 Fusion 界面，则被设想为承载这些 Agent 的新工作台。

那次之后，ONE 在定位上基本开始退场了。它没有立刻消失，但从「钉钉 AI 的答案」退成了下一套答案里的某一层；从主入口退成承载位；从新工作台的想象，逐渐收缩为负一屏。

这样看，ONE 的定位变迁可以概括为几次改写：先在大基数、高频、付费之间找交点；再用工作带发现，试图同时承接 DAU 和商业想象；发布会后回到效率和 AI 组织数，被更明确的指标收束；最后让位给 Fusion，成为更大框架里的过渡层。

每一次改写都有理由。问题在于，前一次留下的缝，并没有在下一次里被补上。大基数和付费深度的矛盾没有消失；工作流和学习流的错位没有消失；看见事和做完事之间的断层没有消失；老钉钉和新入口之间的迁移成本，也没有消失。

定位不是命名，也不是一句对外口径。定位是一次又一次取舍以后，仍然留下来的方向。ONE 最大的遗憾，也即在此：有过很多方向，每个方向都有道理，却没有来得及把其中任何一个方向走到足够深。

设计第三

ONE 的设计，是一个拿着锤子找钉子的方式做产品探索的过程，一个约束收敛的过程。我们从充满想象力的讨论，到放弃一步登天找到全局最优解的幻想，逐渐纠结在一个个看似支离破碎、实则彼此耦合的产品选择中。

这一章，不是为了向读者展示 ONE 当年做过几种卡片、几种交互、几次改版；而是想借 ONE 看清一件更普遍的事：AI 工作产品的设计，不是把模型能力摆上界面，也不是让系统显得更聪明。它首先是在重排人与工作的关系。

谁先被提醒，谁被默认已读，谁需要解释，谁拥有关闭权，谁承担模型误判后的代价，谁能把自己的规则交给系统，谁只能接受系统安排——这些问题看似细小，却决定了一个 AI 产品究竟是在帮人掌握工作，还是帮工作更早地掌握人。

如果读完这一章，读者再看到一个 AI 助手、一条智能摘要、一个自动提醒、一套 Agent 方案时，不只观察它的交互是不是够酷，而会追问「系统里原本这个功能是怎么实现的？AI 做了正功还是负功？用户在获得某种好处的时候，是否付出了更大的隐形成本？AI 如何改变了原本的责任流向？」——那么 ONE 的这段设计复盘，就不只是一个项目的旧账，而有了一点可迁移的价值。

工作内容如何寄身卡片

卡片形态的必要性讨论

余华透露过《活着》的创作历程：动笔以前，他已经「蓄谋已久」地想写一个关于人与自己生命关系的故事，但这种宏大的主题让他苦思冥想、沉吟良久，始终找不到切口，不知道从哪里落笔。

直到某天早上，他一觉醒来，忽然想到「活着」两个字，立刻对太太说：「我知道怎样写这篇小说了，因为我想出了题目，就叫《活着》。」这两个字像一把钥匙，瞬间打开了整个故事。福贵的一生——苦难、荒诞、忍耐、命运——一下子有了一个可以承载它们的容器。

「基于卡片的设计」之于 ONE，就像「活着」这两个字之于余华。

Agent 赛道最大的问题是：叙事强、估值高、但同质化严重，技术护城河非常容易被模型升级吞掉。从技术维度的本质来说，SaaS = 数据库 + 宜人的 UI。钉钉现有的工作内容 AI 化，做成 ONE，是一种围绕既有优势的产品设计。

AI 时代，怎么才算宜人的 UI 呢？

我来 ONE 的时候，ONE 就是基于「卡片」的了。

卡片并非随便选择的容器，它是 ONE 把「主动服务」翻译到具体设计时，最顺手和符合想象的交互语言。

我们在前两章讨论了 ONE 的发心和定位：AI 工作信息流、主动服务、让事找人、钉钉的新首页、用户的专属工作秘书。与之相配的，ONE 的设计方向关键词是「有 AI 感」和「让用户快速消费」。我们希望提供的一种体感是，当用户打开钉钉时，发现系统已经替他多看了一步、多想了一层。



我们希望它不像钉钉原场域，按 IM 消息、日程、待办这样的属类，割据工作内容，按照列表时间排序、红点、DING 这样的形式，组织紧急重要程度排序，让用户在一个个 tab 下进进出出，拼凑零散的工作线索——宏大而泛泛的概念，以至于很难直接变成一个产品，直到「卡片」的概念出现出现——ONE 才有了一个可以寄身的容器：消息、待办、日程、会议、审批、发现，这些原本散在钉钉各处的东西，都可以被 AI 拎出来，排成一张张卡片，主动送到用户面前。

ONE 像一个被收拾好的工作桌面， AI 小秘书敦淑体恤地侧立一旁伺候。当用户准备开始工作的时候，几案上已经恭敬地按序陈列好一本本需要批复的奏折。

这个形态很有诱惑力。

它轻巧直观，适合移动端，而且一下子让人联想到抖音的 feed 流，一款让 C 端用户沉浸上瘾的内容消费形态。

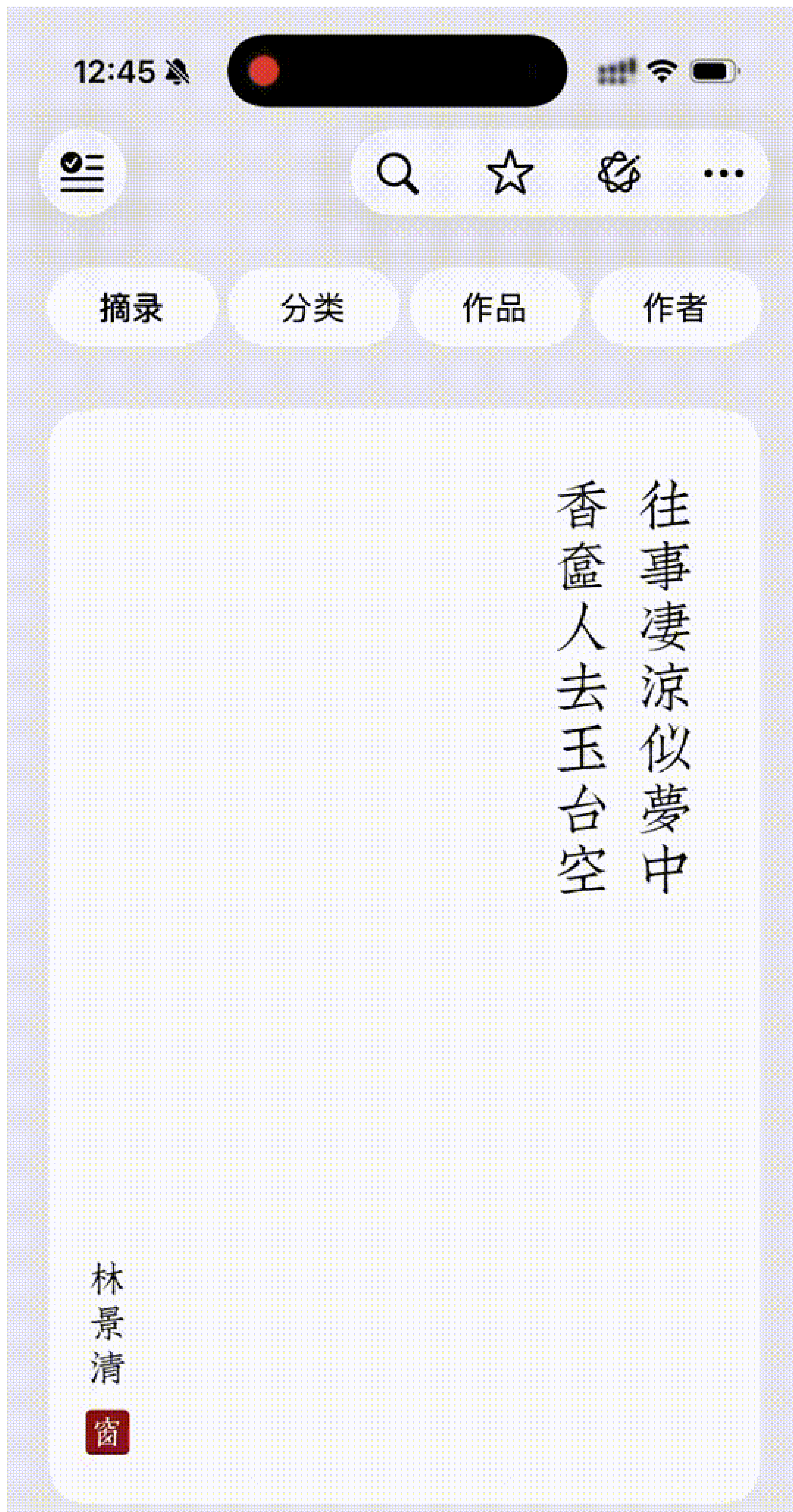
因此，它极宜 demo，尤宜向投资人 pitch。对外沟通客户、开发布会，对内汇报集团，亦然。

卡片设计的风险

不过，因为卡片解决的是「展示」，并没有回答「内核」，走进具体而鸡毛的工作场景：工作并不会因为被做成卡片，就变成可以轻松消费的内容。

这正是 ONE 的设计在历史上许多时刻陷入难产的地方。

请读者与我一起想象一下卡片式的交互。可以联想Tinder、西窗烛这样的内容相对同质的图文卡片，或者抖音这样的全屏 feed 流。


12:45
摘录
分类
作品
作者
往事凄凉似夢中
香奩人去玉台空
林景清
窗



一张全屏幕的卡片送到眼前，用户只需要停留消费卡片上的内容，不需要额外的上下滑动，操作而言，只需要左滑右滑或者少数点击，系统会根据用户的主动标记或停留时长，来记录用户喜欢或不喜欢，并无限给用户推送下一张卡片的内容。

由全屏的视觉展示和极简的操作可以推想，卡片更适合推荐、浏览、轻处理，也即更适合娱乐内容。这个交互操作，落在工作内容上，就会比较操蛋。

工作不是消费了就结束的内容。

划走一张 Tinder 卡片，卡片背后的用户，对你的「不感兴趣」也「不感兴趣」。但是每一张钉钉的工作卡片的背后，都可能有一个紧张等待你表态、回复或者交付的同事。

这不是说卡片完全不行，事实上，有些工作确实适合卡片。

比如审批就很适合。信息字段规整，核心操作单一，同意或拒绝，也正中卡片设计的下怀。待办也适合，因为它能把散落在消息、文档、会议里的信息转成一个待办对象。日程和会议准备，勉强也可以适合——这些场景的共同点是：结构相对清楚，动作相对明确，关系压力相对弱。卡片可以帮用户少看一点、少点几步、少漏一件事。

不过 IM 消息，就是一道天堑了。

IM 消息如何寄身卡片

IM 消息在 ONE 呈现的必要性讨论

IM 是钉钉工作中最重要的一部分，占据用户 80% 以上在钉钉的停留时长。钉钉的文档知识库等知识管理既已积弱，所服务的诸传统企业，本身信息沉淀的诉求又低，IM 的信息就更显重要。

想要用 ONE 给发现带量，再用发现变现，必须培养用户用 ONE 办公的习惯。不带 IM 玩，「高频」「大 DAU」的目标，就流水漂萍了。

落成移动端

ONE 主要落在移动端。

这个选择和卡片的原始设计选择是强关联的。

曾经 PC 难为水

事实上，在 ONE 项目启动的初期，6、7 月份就出过 PC 端的需求设计，里面有涉及钉钉原场域端框架改造的，还拉 PC 端框架的产品和技术负责人讨论过。最终没成，主要因为：

1. 卡片的设计放在桌面端，就显得格格不入，信息量比之前还低；
2. 操作起来更不方便，没法像移动端那样滑屏；



3. 考虑过类 AI 浏览器的方案，也即把钉钉第四屏做成 copilot（现在的悟空入钉也是这样实现的），但是由于没有卡片元素，在提案初期即遭否决。



另外可能还有原因 4.：ONE 是保密项目，我们只被允许不暴露具体产品形态的情况下，兜着圈和「外面」的产品开发讨论方案，对面也是一头雾水，产品讨论起来鸡同鸭讲，你画我猜。

关于项目保密制度，我们会在「秩序第六」章讨论更多，本章还是聚焦前两点设计问题。不过，说起来也还好没有投入开发，不然也是徒然耗人。

再说回移动端，移动端其实不适合直接处理复杂工作：屏幕小，信息承载量有限，只适合提醒和轻量读写处理。它可以提醒用户有哪些工作事项要办，排好顺序，准备好轻量的处理选项，却很难让用户真正「顺手」把这件事完整做完。写文档、填表格、写代码、跨系统查资料、整合总结分发、跟进执行，很多还是更适合在 PC 端完成。

移动端还是 PC 端，这又是一盒薛定谔的用户。只是人力有限，还是要选择一端先下手为强。

可以想象地，我们选了移动端，同时也一直牵挂着 PC 端，记为「重要不紧急」。

和所有「重要不紧急」的任务一样，直到 ONE 死灯灭（其实倒也没有完全死）， PC 端都没真正被排上一天开发。

实时性问题

要在卡片中加入 IM ，就必须考虑实时性问题。

整理复盘资料的时候，发现 7 月底在备忘录上记了一个产品卡点便签如下：

 格式        

2025年7月27日 15:47

731 所有影响可用的卡点

【P0】
++++++++++卡片状态透传++++++++++
1. 卡片重复多张；
2. 已读较久的消息，没有及时杀死（仍然作为未读卡片透出）；
3. 消息已读后，对应红点在IM列表页清除；

++++++++++消息展示交互++++++++++
1. 特殊消息被吞（图片，视频，和部分不支持的消息格式，会被直接过滤掉）（应该发出中括号提示被吞内容，如”[图片]”）；
2. 语音消息无法展示“类型3的消息暂不支持展示”（应该发出语音+转文字）；
3. 文字表情回复严重超时；
4. 【新】封面（极速版）；
5. 【新】消息中快捷添加todo按钮。

【P1】
1. 文字表情一系列交互问题（无法点击撤回表情；长按选中某对话框出现的文字表情不能动态调整，如果已经发出某条表情，再次长按会重复出现该表情且不可交互，etc）；
2. 视觉改版（“前情提要”位置上移）；
3. 字体大小适配。

【P2】
1. Prompts持续优化&评测。

之所以会出现「卡片重复」「卡片上的消息已读，但卡片仍然以未读状态存在」「红点没有消除」的问题，核心皆因这一套卡片是由服务端上异步生产所致。

不实时对 IM 消息卡片来说是致命的。

因为用户消费的对象，80%以上是其收到的消息本身，而不是AI生产的内容。 IM 消息生产频次高、变化快，**不实时就意味着对用户的欺诈。**

如果只是卡片重复读，或者在 ONE 读了一遍、回到原场域又要重读，这还只是浪费用户时间的「小」问题。

而如果是被发信人编辑过的信息，用户在 ONE 里读了一遍编辑前的版本；亦或者是已经在原场域被发信人撤回的敏感信息，又可以在 ONE 的卡片里读到——这些可能就是会涉及数据泄露的大问题了。

其实，如果没有对实时性要求极高的 IM 消息，工作卡片是非常适合由服务端异步生产的。

正如前文提到的 OpenAI 推出的 ChatGPT Pulse，也是夜里异步产生卡片给用户第二天早上起来读。这样由异步服务端生产卡片的好处是：

1. 【生产侧】处理频次低，一次输入输出完整处理，大大节省了成本；
2. 【用户侧】一次性可以读到信息量更高的内容，减少打扰，提高效率；

3. 【产品运营侧】培养周期性进入产品的习惯。

可以说是个多赢。

只是这样，就注定不能满足产品要做成高频的需求——这又回到了定位的不可能三角。

重要性排序

ONE 要主动服务，第一步就是排序。

ONE 针对卡片重要性排序的设计，有过两种思路：

1. 所有工作卡片，不限类型，放在一起打散排序。

   这样做的好处是更加灵活，比如有三张卡片，分别是一条来自主管上级在群里@你的提问消息，一条普通的群聊消息，还有一条还有 1 小时截止的待办。打散排将这条紧急的待办放在重要紧急的主管未读消息之后，和一条不太重要的群聊消息之前。

2. 先规划各种工作卡片类型间的排序，再在每种类型内各自排序。

这样做也有好处：心智更加稳定，也符合心流理论。因为人在不同的任务间切换，认知层面上是需要「重新启动」的：先处理两条消息，再处理一条待办，再处理一条日程，再回来处理两条消息，其实也是对人更加高功耗的要求。

我们最后选了第二种。

由此，实际开发中，ONE 里的排序有两方面，一方面是各种类型卡片间的排序，这其实也就是一套人工规则；另一方面是 IM 消息的排序，这里面的复杂度就要高很多。

（其他卡片类型内的排序也是非常简单的一句话人工规则，比如日程只和时间有关，待办也和截止时间有关。）

各种类型卡片的数量级，和用户在对应原场域的工作内容量，是成比例的。IM 消息占到钉钉用户工作的 80% 以上，对应的，用户每天卡片数量的大头也都出在 IM 卡片上。

排序的设计思路上，我们主要根据组织关系（比如直属上级，重要联系人）、信息性质（比如 @ 我、DING消息）；行为亲密度（比如会话曝光点击、发言内容长度、停留时长）等组成的上下文，来构建会话的紧急重要排序。说到底，是想让系统理解企业消息里，对一个用户个性化的真实重要性评估。

卡点 1： 排序和分组的矛盾

有一些 IM 原场域的特性，如果不拆碎了重整，是非常难塞进卡片排序这套逻辑里的，其中最典型的特性就是分组。

一般什么样的用户会使用分组？首先，会话足够多，消息足够多，否则没必要分。其次，它在处理一些不同领域或不同角色身份的会话消息，否则也没必要分。所以概括起来，负责多个项目的信息中枢型员工更可能使用分组。这部分用户，也符合我们在用户定位里讨论的塔尖用户类型。

对应的，也有两种用户常用以分组的手段。以 PM 为例，同时在对接许多项目，也同时在对许多不同的岗位角色沟通。这里就有两种分组思路，一种是按照项目分组，另一种是按对客户、对内部这样沟通去做分组。

当用户在原场域切换分组的时候，页面上就会把这个分组下所有的会话都陈列出来，不区分已读未读，不区分重要程度。

如果要做排序，那么必然未读在前——甚至只能放未读，否则是对有限卡片空间的浪费。同时，各个分组内一定会被打散：一个不重要的分组里的客户愤怒的紧急舆情质问，很有可能重要过一个重要的分组里某个内部同事的玩笑发言。

所以要把分组也搬进排序卡片里，天然存在着不可调和的矛盾。我们从具象的例子走到抽象的概念，主要是这些维度上的不可调和性：

维度	分组	排序
核心目标	提供稳定的语义归属（谁在说什么话）	提供动态的行动优先级（我该先看谁）
稳定性	相对长期稳定（项目、角色、主题）	高频波动（随时间、未读状态、紧急程度变化）
用户心理	「我知道它属于哪件事」	「我现在该先处理哪件事」
弊端	可能掩盖紧急但不常用的对话	可能打乱用户的熟悉结构

我们实时同步处理IM消息的初衷，以及导航栏滑切的交互，决定了我们一定以「排序」为主要组织维度，「分组」作为辅助。

或者说，分组应该作为排序的加权项，而不能完全照搬。

我们提出过用标签形式软化和替代分组的可能性。不过这个方案最终没有通过。最后老板拍板还是希望我们把分组完整地搬进 one，最后的呈现形式也比较怪异，就是单拎了一个分组的卡片类型出来，卡片上放分组消息列表，当出现未读的时候呈现整个列表。

可以想象这是非常别扭的。首先，对于一些有大量分组，或者分组已经老化的用户来说，这样显著降低了卡片消费效率，其次，对于一些用户同一个会话设定在不同分组的情况，就更加难缠。

这个功能做上线之后，我们项目室里就有开发拿着手机找我展示他的分组消息卡片，每张卡片上只有一两个未读会话，零零散散分了十几张卡片。他苦笑说，幽素有没有办法调整一下？这是 bug 吧。

我说...可能没办法，这是他要的。

说这话的时候，我自己也脸红，感觉自己在逃避和推卸责任，不是一个好产品。

卡点 2： 排序被卡片交互遮住

排序要成立的一个前提是，首先用户必须感知到排序。

列表天然让人感知顺序。第一条、第二条、第三条摆在一起，用户会比较，也会逐渐相信这个系统把重要的东西排在前面。

卡片不一样，卡片一次只出现一张。用户面对的不是一个序列，而是眼前这件事。即便再把卡片放进导航栏、横滑、头像和各种压缩空间里，用户首先感受到的，往往不是「系统帮我排好了」，而是「这个会话我看不清」「这个人是谁」「这条消息是不是已经被我读了」。智能没有消失，只是被交互遮住了。

如果眼前这件事还是未读的重要消息，用户就更难感知排序的好处。

这也是早期消息卡片很别扭的地方。系统花了很多力气判断重要性，用户的第一反应却不是「它排得准」，而是「它会不会替我暴露状态」。排序在卡片里被压力淹没了。

所以，排序是主动服务的起点，但排序必须被合适的界面承接。否则系统判断得再多，用户也不一定感受到价值。

比如，有没有可能直接做进 IM 列表里呢?

可惜，那是 ONE 的命题以外的工作了。我们尝试过「向前一步」，汇报提议过这个方案，但是可以想象地没有任何结果。

此外，相比「难以感知排序的好处」这种隐性 debuff，还有一个显著高亮的debuff：用户的注意力会完全被另一个问题攫取——

已读恐怖主义

「卧槽，这怎么直接把我已读了？！」

无论是内部同事还是外部用户，第一次在 ONE 的卡片里刷到 IM 消息，几乎都是这个反应。

消息卡片里最棘手的问题，不是排序，而是已读未读状态。

已读未读，其实也就是消息阅读状态功能，最早可以追溯到2005年，由黑莓公司在 BlackBerry Messenger (BBM) 中首创，设计初衷是为提升商务沟通效率。钉钉 IM 消息接入已读未读显示，是钉钉在 2016 年 1 月 16 日推出的版本 1.0 里的当家得意之作。

已读，即开启了回复的倒计时。

这在钉钉原场域，是用户早已习惯的机制，但在 ONE 在的卡片里却是大灾难。因为用户不再是观察未读消息列表，通过 last message 简单判断消息内容，再主动选择一条进入，清除已读状态，而是直接开了 AI 秘书呈上来的盲盒。

剥夺了用户的选择权，责任和后果却要用户承担。这委屈谁受得了?

那能不能在 ONE 里就不直接帮用户已读掉呢?

内部一开始讨论过其他方案，比如「只读卡片不算已读」「进入 IM 详情页才算已读」。这些方案都被否决了。无招认为，这会损害发信人的利益，也会动摇钉钉的根本。这里主要是发信人立场，可能也有一点产品一致性的考虑。

更糟糕的是，这里的已读，纯粹是锦衣夜行。

过去钉钉做已读未读成功。对比的是微信。用户一下子就有体感了：是钉钉帮我，让我布置出去的信息都可以查看状态，让我对一件事情的进展有更强的把控力。

然而在 ONE 里，现在收信人是清楚自己在这儿半推半就、将信将疑地已读了几条他还没准备好签收的工作任务。而与此同时，对面的发信人，其实根本分不清自己发的消息，对面究竟是在钉钉原场域还是在 ONE 已读了自己的消息。于是，我们千辛万苦硬着头皮做出来的已读，不仅吃收信人的瓜落，发信人也不买它的账。

后续「用户」章也会提到，我们收到针对卡片形态最强烈的反馈也落在这里：「我怕客户/老板看到我已读不回。」

有时候，不立刻点开消息，不是怠惰拖延，而只是给自己留下一点处理的余地。

一个对自己工作内容熟稔于心的工作者，从群名、发信人和 last message 的上下文里，已经可以大概推知对方问什么要什么了。

一个运营同事反馈过类似场景：她看到是哪个群里、哪个人 @ 了她，就知道大概在问哪件事；但她还没准备好回。比如某个不是非常紧急需要的回复，可能依赖一个小时后要开的会议的结论，那么等会后有了结论再同步对方，就可以一次说清。如果现在读了，就要多解释一轮，多发一句「我看到了，但要等几点几点某某会后同步」。

ONE 的消息卡片把这点余地压缩了。它原本想替用户减少读信压力，实际却可能提前启动责任压力。

更尴尬的是，这个设计对发信人的收益也很弱。发信人并不知道用户是在 ONE 里看到的，还是在会话里看到的。他只看到一个「已读」。对他来说，体感差别并不明显。

所以这不是抽象的「发信人 vs 收信人」立场之争，而是一笔具体的收益和成本错配：发信人获得的增量很弱，收信人承担的不安很强。

我后来越来越觉得，这是一个锦衣夜行的设计。它维护了一个抽象正确的立场，但真正被保护的人没有明显感知，真正付代价的人却清清楚楚。

当底层规则不能动，交互就只能补洞。补得再巧，也还是洞。为了缓解已读带来的压力，我们后来尝试过一些补丁。

无望的补丁 1：横滑

8 月 25 日发布会前后，ONE 引入了横滑结构，从变成了 Tinder 进化为微信公众号。


5:26
工作
网钉科技有限公司
1
直属上级
2
@我
11
刘悦
特别关心
11
特别关心
5月28日 周三
TODAY
Hello！以下是你今日需要关注的所有事项，请查收~
日程 4
待办 3
审批 4
59秒后开始
智能化产品联合验收
09:00–10:00
C6–4–04N
宇
6
未响应，1个日程冲突
智能化产品运营方案
09:00–10:00
项目室



它本质上还是卡片，只是在页眉加了一排头像。这样用户能提前知道后面大概是谁的消息，也可以点击跳过某些卡片。

横滑结构设计还是想解决两个问题：

1. 给用户预告感，缓解「不知道下一张是谁、但看见就已读」的压力；
2. 允许点击头像，主动跳过一些卡片。

然而，交互的底层规则没变，另外还增加了两重隐形而严重的麻烦：

1. 相比原场域有 last message、有消息发出时间等，用户在 ONE 需要依靠更少的信息——仅通过头像判断可能需要跳过谁的信息；
2. 页眉加了横滑头像之后，压缩了可读空间。在一些小屏幕机型上，键盘一拉起来，甚至看不到 IM 消息本身，反而更加损失了信息效率，加重了用户的阅读负担。

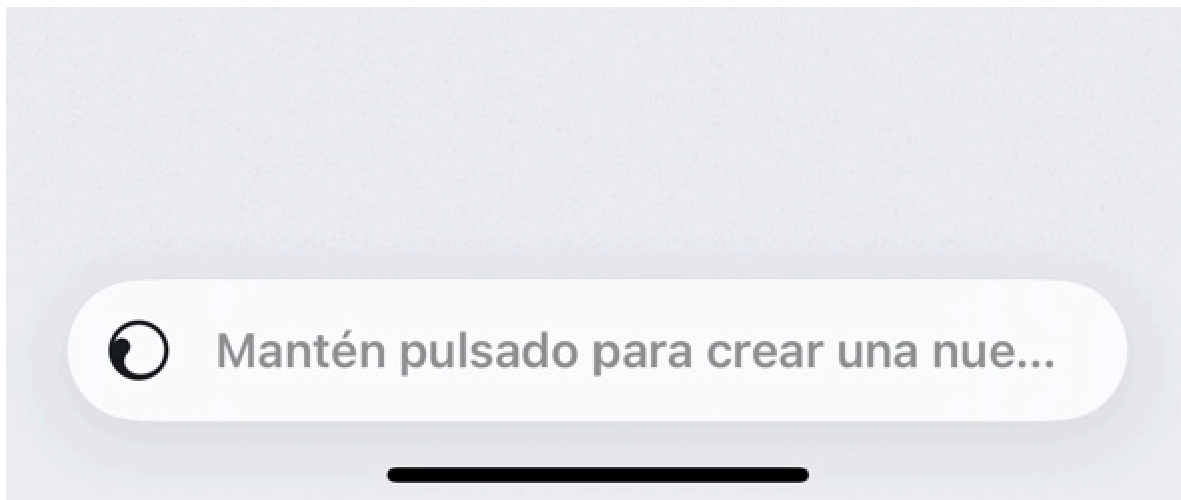
横滑相比原场域的好处，仅仅是减少了用户一进一出、点进一条消息查看的成本，但这对用户来说真的重要吗？假设点进点出需要耗费用户 1 秒，而判断一条消息需不需要跳过需要 10 秒。为了节省 1 秒而增加 10 秒的负担，这真的划算吗？

无望的补丁2：Peekaboo

Peekaboo 是我提出的。它的原理是，既然必须要标记用户已读，那么我们可以在明确标记已读的节点上略施花活。

用户在横滑到下一张卡片时，只要没有完全划到下一张、没有松手，就还不算正式读到，可以理解为「预读」。这样用户可以比普通 IM 列表里的 last message 看到更多消息内容，又暂时不触发已读。


12:29
5G
幽素
钉钉
27
mole...
Más Ch...
事项助理
任务助理
Explorar
Clear Badges
AI–Minimize
箭派
13m ago
晓童)：欢迎阿迪加入本群
州租房群（公...
19m ago
)：150出蚂蚁月饼，不包邮
1
Daily On
自动化小助
6
A1员工内
Big(炎御)：



这个设计纯粹是无奈的、苦中作乐的、抱有侥幸心理的、但好在也没花什么成本的——补丁包。也如预料，几乎没有用户主动发现这出精致的淘气。

卡片的可玩性

当然，卡片绝非一无是处——如果真是一无是处，我们也不会启动了。

除了可以减少用户一进一出的操作，以及节省判断时间之外，卡片的想象空间很大，或者说可玩性很强。

卡片格式可以承载非常多的内容，比如我们做过「今日消息今日毕」「今日金句」等有意思的功能：

5:25 

工作 1 

今日金句

· ✦ perspective ✦ ·

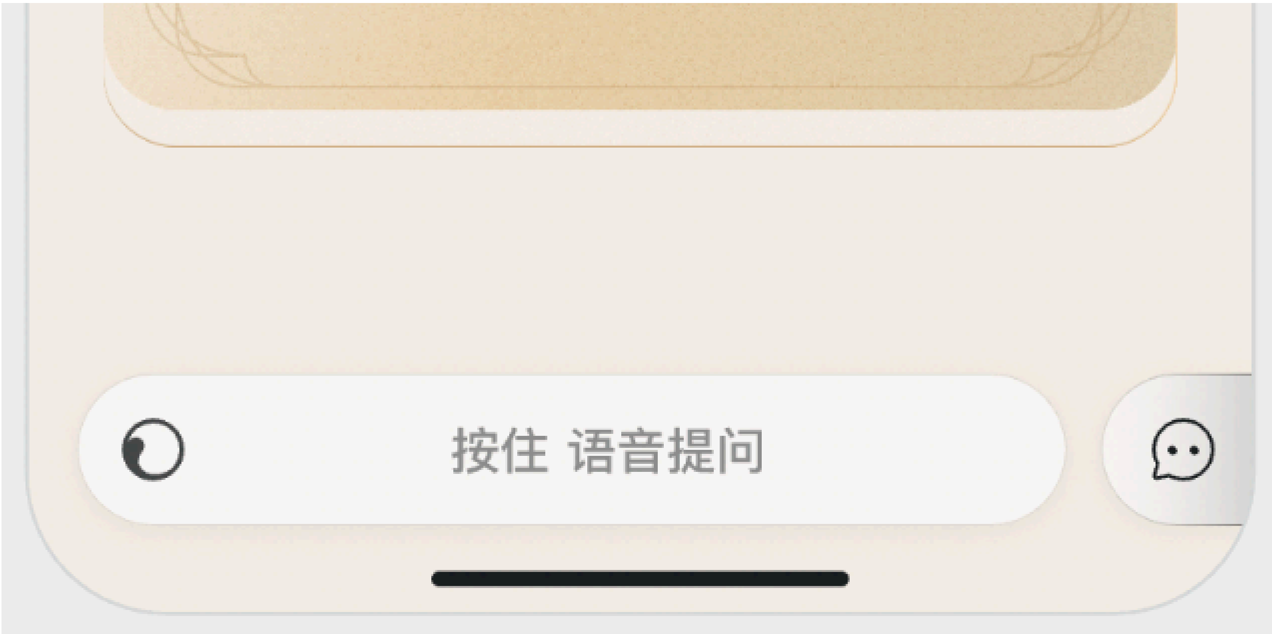
“

人类彻底成为驾驶员

以后只需要外包啦

我要强调一下我一直说的：迟早要被大模型创死 逃避没有意义 不如让🔥烧的更旺一些

 xxx · 来自「AI摸鱼社区」群 ›



他们的定位是消息的解构和重新组织，辅助用户快速消费信息或者重吸收。

对标原场域

2.3 节提到的实时性问题，其实就是对标原场域的一个细分场景。

讨论和走查对标原场域，其实就是在计算用户的迁移成本。

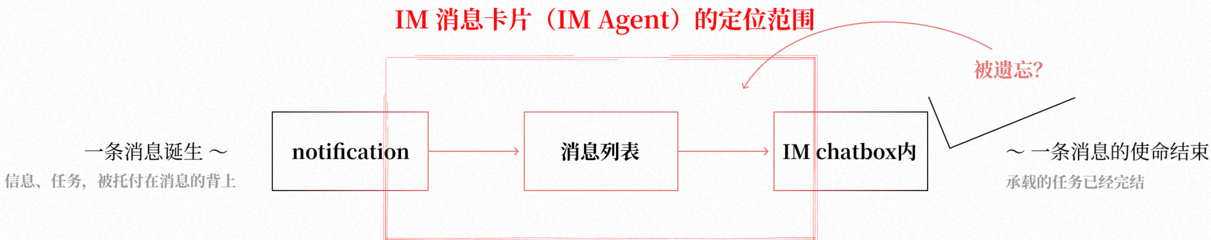
如果说，我们的最终目标是用户可以完全迁移到 ONE 上，那么IM 消息卡片究竟对标和替代原场域什么功能的职能呢?

首先，替代了一部分阅读消息本身内容的职能；但是并不是所有消息都适合在移动端展示呈现，如果涉及长消息、文档、链接、动态卡片，还是更适合在已经迭代了十年的原场域或者回到 PC 端查收。

其次，替代了一部分提醒通知。我们一开始讨论希望 ONE 里读了消息可以算作「未读」，就是考虑 ONE 可以对标 notification bar的可能性。这也很符合常理：毕竟卡片可展示空间有限，可以认为是一条长提醒。

第三，鉴于我们做了「已读未回」「今日金句」等卡片，相当于帮用户做消息的「重吸收」，因此我们也可以对标这样一部分超越原场域范畴的工作。

最后，消息列表。IM 消息卡片的目标，可能是要把整个消息列表吞掉。这并不是因为我们已经对标了 IM 列表的能力，而是我们在卡片的形态里，已经没有列表的容身之处了。



不过，我们真能替代得了么?

以 IM 消息列表为例，它不仅承担未读按时间排序的展示，它实际上容纳了所有的会话。当用户想要根据一些模糊的记忆，对照时间线模糊下翻寻找某个会话的时候，当用户想根据 last message 大致预知未读消息中可能包含内容的时候，当用户长按某个会话，把它标记为未读的时候——还有许多其他场景，IM 列表其实隐形地承担了非常多的职责。

原场域模块	能力拆解	ONE对标模块	ONE中情况	严重程度	决策
IM列表	按【处理时序】模糊查找会话	导航栏	🈚	高	
	按【分组】查找/管理会话		仅透出含未读的分组，以列表形式呈现；无主动查找路径	高	
	last_msg，【预告未读】信息原文（~20字）		🈚	高	
	长按出会话快捷操作		🈚	低	
	last_msg，【预告未读】标签		有[有人@我][直属上级*][特别关注][新会话][草稿][稍后再看]，没有[@所有人][直播][红包]etc	低	
	last_msg，【监测对方已读】情况		🈚	中	
	last_msg，时间		🈚	中	
	对方【状态】，如“会议中”和自定义签名		🈚	中	1
	对方【组织】，如“@xxxx公司”		🈚	中	
	对方【群性质】，如“外部群”“答疑”		🈚	低	
	常驻入口：【置顶】会话/专注模式Pin住会话		🈚	低	

另外，还有一些原场域的隐性优势我们是缺失的，比如说消息可视区域的空间。为了。缓解已读的恐怖，我们做了横滑和页眉的头像，这是一块信息浓度很低，又很占空间的区域。

聊天页	可视区域大小：580 拉起键盘后可视区域大小：235	消息详情页	导航栏大头像:478 小头像：575 • **拉起键盘后可视区域大小：** ◦ **大头像：72 -168** ◦ 小头像：	高	1

讨论卡片形态的时候虽然兴奋，在实际推进中，由于我们把重兵压在了移动端的卡片流，产品展现出来的形态，只是把这些信息提炼成了几行简短的摘要卡片。

这导致了最致命的错位：AI 在台面下做了最难的「网状逻辑缝合」，但在台面上，却只给用户呈现了一个「老钉钉里用肉眼扫一下也能看到」的简短摘要。 用户感知不到那个「不依靠 AI 就无法解决的降维打击」，只看到

了一个多出来的、需要大拇指去滑动的卡片。

新范式的高昂信任成本（用户需要担心 AI 摘要得准不准、会不会漏掉重要信息），最终败走钉钉原场域那虽然痛苦、但高度产生安全感的「手动刷红点」的肌肉记忆。

成本问题

如果说，上一节「2.7 对标原场域」是我们在计算用户的迁移成本，那么由于功能设计涉及 AI，即必有 tokens 的消耗，我们也需要计算这个 AI 秘书日常运行的成本。

之前做成服务端方案时，没有考虑到实时性，成本还相对可控。要做成实施排序的方案生产出来的卡片用户却不一定消费，岂不是「损成本以奉功能完整，犹割股以啖腹，腹饱而身毙」？

格式

2025年8月10日 16:49

[反思] 实时性数据 * AI

【消息 AGENT】<u>**用户消费对象**</u>：80%+是消费**消息数据本身**，而**不是 AI 生产的内容**。

数据	**AI 生产物**	**后果**
不实时	不实时	用户消费不到想要的东西，生气离场，**产品死掉**。
实时	实时	成本超级高，入不敷出，**产品死掉**。
尽可能实时	不实时	唯一的路，先活下去再说。

但是不实时是不行的。上文已经讨论过，不实施就是欺诈。我只能在心里悲壮地想：「而今不实时死，实时亦死，等死，死实时可乎？」

不过办法总是人想出来的，办法总比困难多（不包括 2.4.1 的分组问题）。后来我们用了端上的小模型，做了实时和离线计算的组合，从某种意义上解决了这个问题。

那么，IM 卡片最终服务好它的目标用户了吗?

显然是没有的。

除了上述提到的所有问题，有一只房间里的大象至今尚未讨论，那就是：

卡片整体是用来阅读，而不是操作，或者用来查找的。

钉钉这次大改版的逻辑是什么？硬创新？ 收藏

人气[24353] 回复[179] 5字以上回复 | 时间 热度

闫洪磊(三寻)
工号：107328
部门：云智能集团
芝麻：1093

楼主 发表于：2025-12-18 09:02 只看该作者

针对这次钉钉 UI大改版， N多人被当做小白鼠在多套 UI桶内测试，内网吐槽不少了，但是至今没有看到新版产品设计理念的解释，为什么要改动已经习惯了的工作方式，发消息打电话这么高频的场景为什么隐藏这么深？？？

因为功能太多？假如你真的要归类那是不是可以让我们自定义哪些是要收缩那些要固定？

假如（我说是假如）新版 UI一定要做下去，那么建议方案：常用操作可以自定义，例如下图中绿色区域可以自定义一些自己认为的高频场景，按照我们的工作方式应该会把【视频会议】固定在绿色区域内

既然已经当了一段时间小白鼠了，那么钉钉 PD是不是有足够的数据分析这个改版是不是要再继续下去？还是会听从用户的反馈，不要再强迫更改客户习惯而是提供新的 UI的同时支持自定义更加符合个人工作模式！

参考一下微信，这么多年了对于发消息打电话这个使用习惯改变过吗？

我们在 2.5 节，站在发信人的立场，挣扎着做了「已读」的操作，却没在发信人面前讨到好；我们在 2.7 节走查了和原场域对齐还相差哪些内容——到这里，卡片对发信人来说最致命的问题立刻浮现了：

消息列表承载的查找会话、管理会话的职能在卡片形态上是完全丢失的，我们做了语音指令能力来填补搜索查找，另外在卡片上加入搜索图标来缓解前者，但是效率已经大大降低。读者可以想象一下，在脑海里列出，你最近几天高频沟通的 top 10 群聊的具体名称。由于平时我们不需要输出这个名称，所以直接输出是困难的，尽管你可以在看到它的名称时一眼认出，那就是它。

同时，在 IM 列表上找到这个高频会话只需要 2～3 秒，如果你把它设成置顶，你可以凭手感在 1 秒内找到它。但在卡片， 15 秒是一个非常保守的估计。

1. ONE中[搜索-发消息链路]

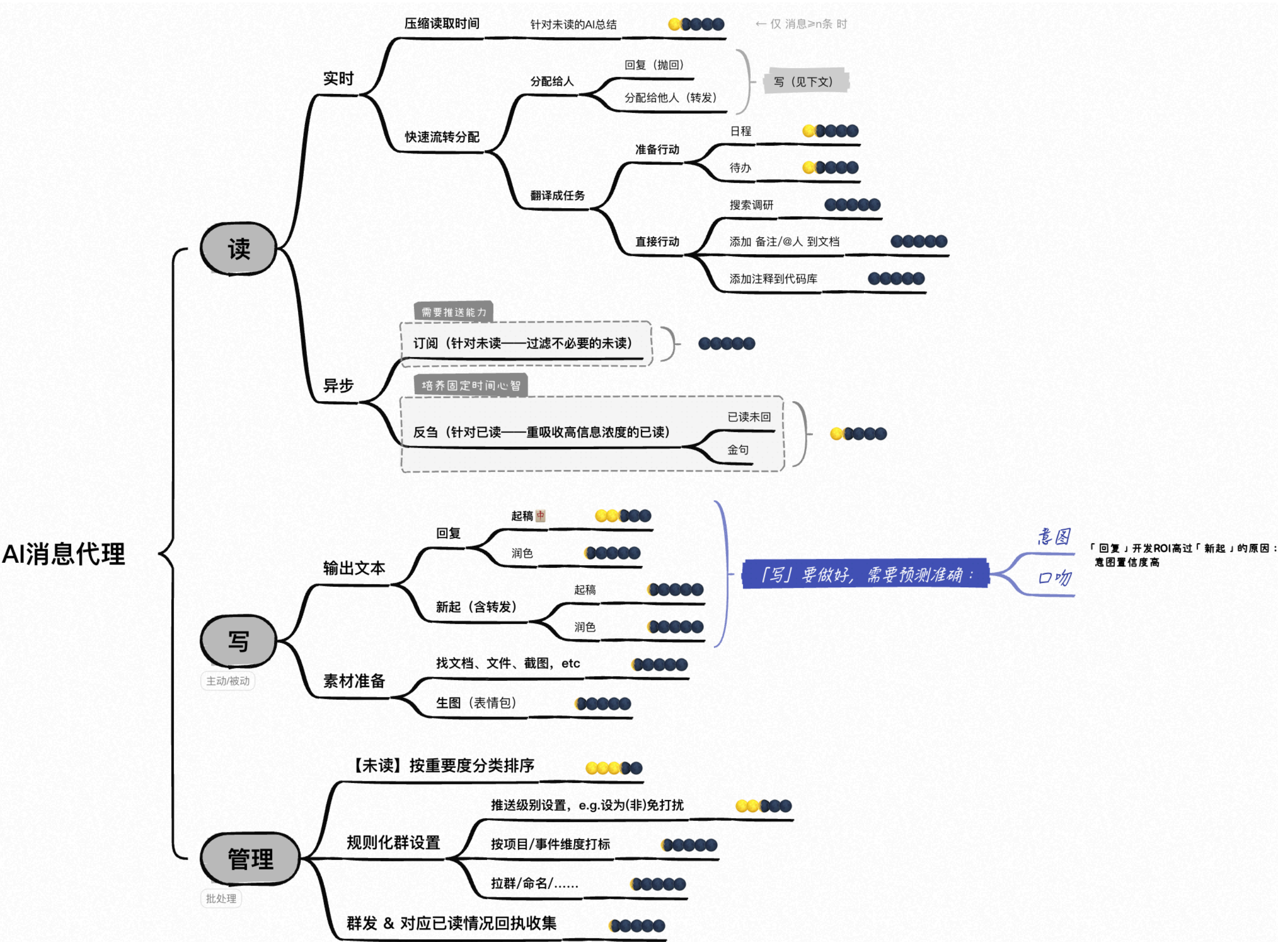
目标：发消息链路在ONE中完成小闭环。
方案：

步骤	Step1. 开启搜索	(Step2. 输入)	Step3. 点选会话	Step4. 进入会话，发消息
示意图	10:50 Work IM Agent All processed Hold to create a new to-do	10:52 Search All Contacts Group Chats Chat Records Guess you want Knowledge AI	10:52 scrum All Contacts Group Chats Chat Records You may want to find O项目一方 Agent(25) O项目一方 Agent Scrum沟通 (15) O项目 IM Native 化Scrum(10) 试试 AI问答：scrum	10:52 O项目一方 Agent [日程 Agent] 全量标准 钉钉文档 Editable Add To Please enter...
逻辑	【前置步骤】点击右上角搜索框	【前置步骤】查找会话（如有）	【前置步骤】单击选中会话	进入ONE消息页。 if ONE进程中有该会话：

重新拆题：AI 到底能帮用户读、写、管什么

到 10 月中旬，团队已经围绕卡片、排序、已读折腾了一大圈，我逐渐强烈意识到一件事：或许设计不该先从卡片开始，而该先从用户动作开始。

用户在原场域究竟在做什么？核心动作是在读、写、和管理消息（分组、打标签）。



于是我们可以把 IM Copilot 的目标重新写成三件事：

1. 提高用户读信效率，
2. 降低用户写信门槛，
3. 兜底用户丢信风险。

寻找甜区

寻找甜区的思路，主要是找到用户有体感、会出现 Aha moment 的，和设计开发成本低的交集。

	可以出Aha Moment	没感觉
设计开发成本低	**【回复】起稿** 优化中 【新起】具体场景起稿：客户沟通，社交发起，进度追问(某种意义上仍是回复) 【新起】【转发】选对人 + 起稿 设计中 **【素材准备】找文档-近期文档** 设计中 **【翻译成任务】模糊跟进待办（e.g.“下周/年后”——“提醒我跟进”）** 设计中 【翻译成任务】搜索调研 **【管理】设为免打扰** 开发中	【读】未读总结（仅基础支持） 【读】“金句” 【读】已读未回（不痛不痒型） 【回复】润色 【新起】具体场景起稿：年节祝福 【翻译成任务】明确具体的日程 【翻译成任务】字面意义的待办 【管理】未读重要度排序
设计开发成本高	**【读】已读未回-命中** **【读】消息订阅** 可尝试小模型 【读】支持秘书语音播报等特殊场景(「听」消息) 【转发】选对上下文消息 可尝试小模型 【素材准备】找文档-远期文档 or 依赖知识库 【素材准备】生图（表情包） 【翻译成任务】待协调的日程 【翻译成任务】添加 备注/@人 到文档 【翻译成任务】添加注释到代码库 【管理】按事件维度打标 【管理】未读重要度排序	【读】未读总结（精细优化） 【管理】拉群 【管理】群发

遵循这条思路，我们在 ONE 里做了 AI 回复。它后来取得了比较好的反馈，并且并入了钉钉主端，成为了灵动回复。

默认值的权力：平台产品不能只靠主厨审美

AI 工作产品里的默认值很重要。

默认是否已读，默认展示什么卡片，默认排序规则，默认是否出现发现，默认是否主动推送，默认把用户带向哪里。这些看起来是产品设置，实际上是产品价值观。大多数用户不会一项项配置，也不会花时间研究开关。默认值就是他们实际生活在里面的制度。

ONE 的默认值里，带着很强的产品设计者的意志。

比如默认读到消息即已读，默认系统替你排序，默认工作卡片后面跟着发现，默认让用户先进入一个由系统安排好的信息流。对于一个小众产品来说，这种强风格可以成立。

像番茄钟一类产品，本身未必解决多深的痛点，但它提供交互美学、节奏感和仪式感。喜欢的人来，不喜欢的人走，它像一家餐馆，有自己的菜品、审美和规矩。

ONE 不一样。

ONE 想做大 DAU，想做钉钉的新入口，想覆盖广大基数用户。它更像平台，不像餐馆。平台产品当然也需要风格，但它不能要求所有人都按主厨的口味吃饭。一个小餐馆可以坚持「我就这样出品」，美团不行。美团要承载千千万万种吃法。

ONE 的许多设计问题，正在这里。它想做平台级入口，却采用了很强的主厨式默认。老板和少数产品设计者反复耳提面命，把自己的工作方式、审美、判断，做成了所有人的默认处境。

发现是典型例子。

发现不属于工作卡片，更像学习流。里面会出现 paper、学习内容、视频化内容。某种意义上，它是一个被放进工作入口里的内容频道。

发现的存在有几重原因：无招个人对学习内容的执念，商业化压力，以及一种组合思路——工作模块负责高频和 DAU，发现模块负责知识订阅付费和 token 消耗。三者并不矛盾，甚至可以互相解释。

问题在于用户场景。用户打开 ONE，主要是处理工作。消息、日程、待办、会议都是压在眼前的事。刷完工作卡片之后，如果系统自动进入发现，开始播视频，几乎所有的用户都会被吓一跳。很多反馈也很直接：「像广

告」「工作时不想看」「占地方」。后台和 ONE 用户反馈群里，要求把发现关掉，是非常高频的问题。

几乎没有普通用户给发现正反馈。

有少数老板或 KP 会喜欢，比如市场部 KP 希望用来监控舆情，但他们也更多希望自己用，并不会给全体员工开权限。不过我们对发现的期待，是老板能给全体员工开；不然这要怎么（足够高地）收费呢？

这又回到平台和餐馆的问题。少数高能动用户可能喜欢学习内容，喜欢信息摄入，喜欢随时看到外部变化；多数打工人打开 ONE，只想把今天的工作处理掉。发现的问题不在内容质量，而在时机和默认。学习当然有价值，可当一个人正在处理消息、会议、待办时，系统突然递来一个学习流，就很容易从服务变成岔路。

从这个角度出发，也再一次印证了用户自定义的重要性。越有主观能动性的用户，越希望自己能调规则：我想看哪些人，哪些群不要提醒，发现能不能关掉，什么消息算重要，哪些卡片以后少推。可 ONE 早期给用户的自定义口很弱，更多让用户接受系统安排，而不是让用户驯化系统。

探索非卡片的可能性：信息流

上文提到过和展示过西窗烛的划卡片式交互。

其实有两个非常有名的古诗文类 App，还有一个叫古诗文网。古诗文网就是一个典型的无限信息流式的首页，和今日头条非常相似：所有呈现的信息可以在一个页面上，可以向下一划再划，但是不会提供像卡片那样切割出来的、一件事是一件事的手感。

12:45

搜索 畅听

足迹 推薦 詩文 名句 古籍 作者

高中古詩
GAOZHONGGUSHI

李賀 唐代 译 注 赏 音

馬詩二十三首・其九

飂叔去匆匆，如今不豢龍。
夜來霜壓棧，駿骨折西風。

寫馬 憤慨 懷才不遇 隱喻 寓事

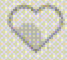
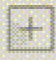
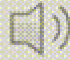
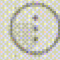
李清照 宋代 译 注 赏 音

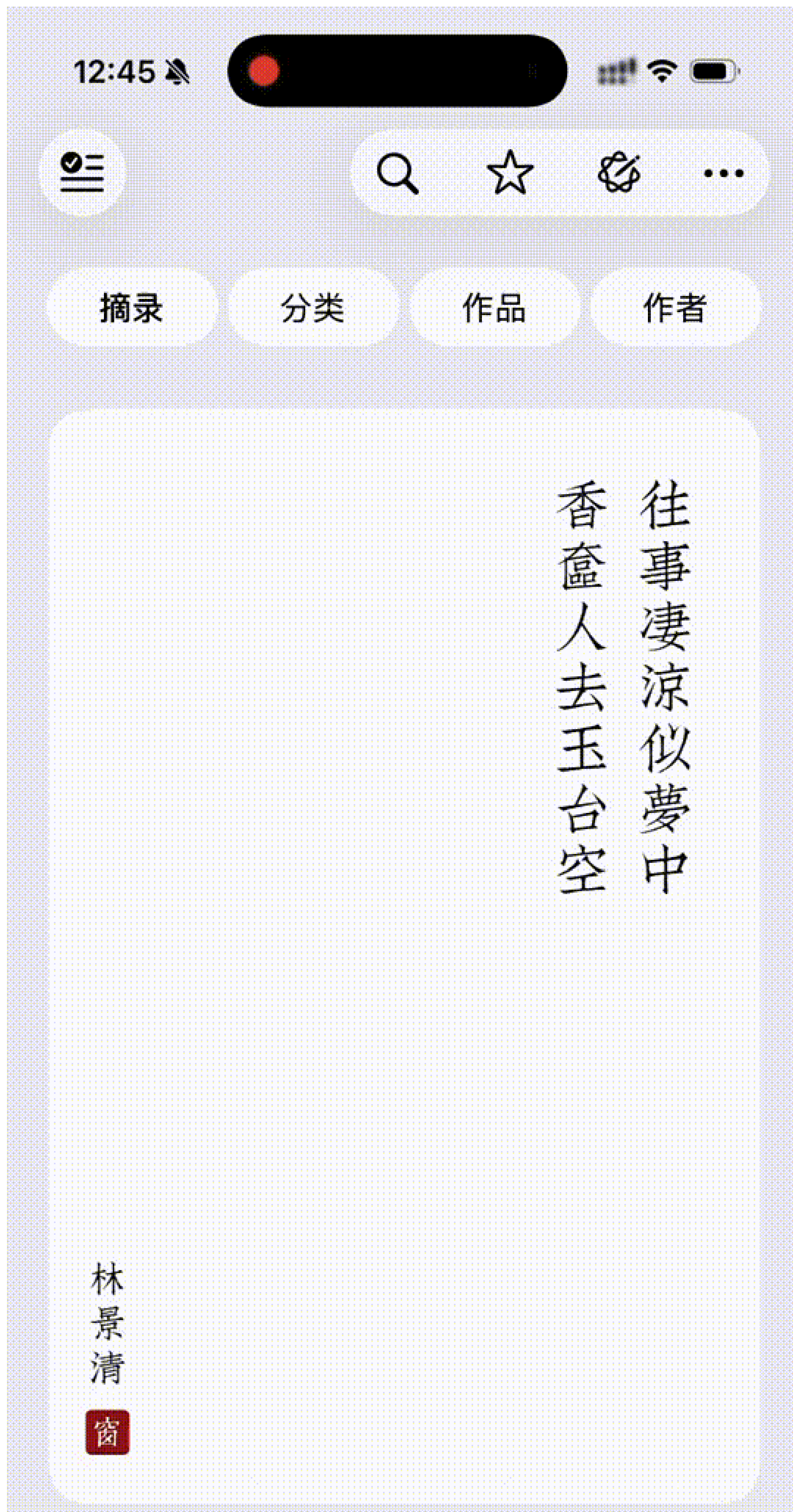
訴衷情・夜來沉醉卸妝遲

夜來沉醉卸妝遲，梅萼插殘枝。酒醒熏
春睡，夢遠不成歸。

VIP 朗读可选择

人悄悄，月依依，翠簾垂。更接殘蕊，更捻餘香，更得些時。

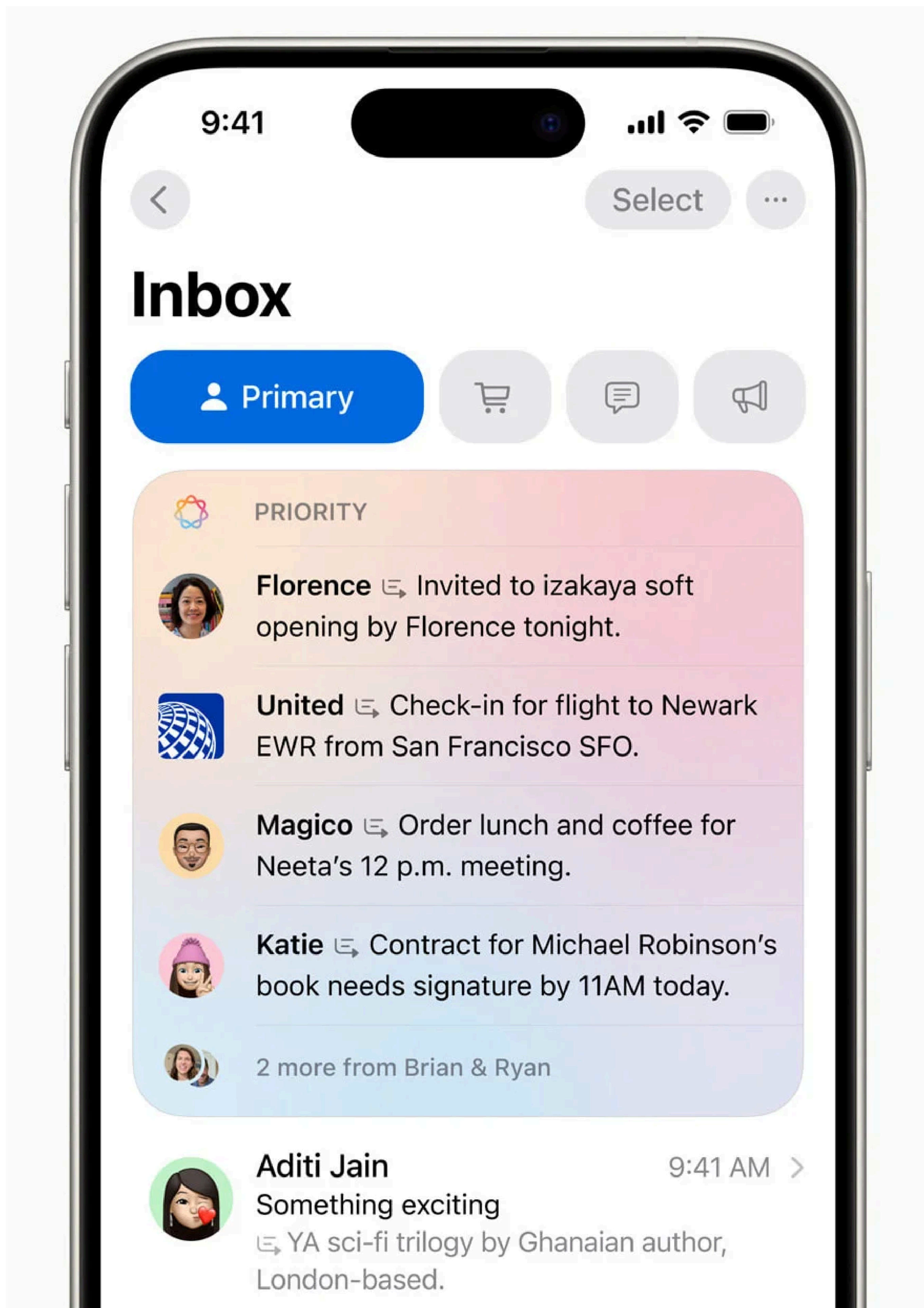
    


12:45
摘录
分类
作品
作者
往事凄凉似夢中
香奩人去玉台空
林景清
窗



在 ONE 走出项目室、产研团队也收缩为一个更小规模团队的时候，这种可能性终于被我们提上日程。

我们的新目标是，在一屏内展示所有重要的内容。首先，我们重新把重要未读，以列表的形放出来了，接着我们在下面同一页内，继续呈现当天重要的日程、需要做的审批和待办等等。



这个形态还是受到了一些好评，尽管它失去了最初项目启动时理想主义的卡片手感。

放弃大包大揽

其实到后期，放弃「什么都想要」的心态，不做满手抓着金银宝翠一颗都不肯放手的强盗，反而好了。只抓重点一个心智：当首页为空时，用户即已解除当前所有紧急重要的依赖，不再是任何人的卡点。



上了这个版本之后，在我们没有任何主动放量和宣传的情况下，产品还是保持比较温和的自然增长。比较令人惊喜的是留存，次流从 10% 出头涨到接近 30%。到 one 入口被换成悟空的 4 月份之前，次日留存涨到巅峰 45% 以上。

小结：形式不该凌驾于内容

回头看，ONE 的设计教训不在于卡片不能做。任务识别、会议准备、快速审批、AI 回复都证明，卡片和 AI 可以很好地承接某些工作。真正的问题在于，工作产品里的「看见」从来不是中性的。

看见一条消息，可能意味着已读；看见一个待办，可能意味着责任；看见一场会议，可能意味着准备；看见一个总结，可能意味着后续分发。系统把事情提前端到用户面前，不能只计算阅读效率，还要计算责任成本。

设计也不只是把界面做好看，而是安排权力和责任怎么流动。

AI 工作产品的好设计，不是让系统更早发现工作，而是让用户更好地掌握工作。

如果没有做到这一点，所谓的主动服务，只是变成对用户傲慢而爹味的控制欲。

用户第四

其实在我的理解和倾向上，「用户」和「设计」之间，「用户」排序应该更靠前。用户是靶子，产品是箭；焉有靶子在哪还不清楚、甚至场地和比赛项目都没确定，就先挑选箭支的道理呢？

然而行文顺序最终定下是「设计第三」「用户第四」——因为就实际执行而言，ONE的开发过程确实有很多「设计」在前，而拿「用户」来贴近设计的成分。

这不是钉钉独有的问题。

无论是做 AI 产品验证 idea，做其他产品，或者泛化为做一切实验假设，一旦立了项，或做出了某种初步假设，沉没成本就会让这个产品或者实验假设的发起人，下意识成为自己假设的拥趸。一个人提出一个念头，很快就会开始保护这个念头；一个团队做出一个形态，很快就会开始解释这个形态；一个组织投入了资源、时间、排期、汇报链路，就更难承认：也许一开始题就没问对。

大型组织的沉没成本更高，因而更容易陷入自恋、自我解释和难以掉头。

如果说前三章更多还是是逻辑学科，那么从本章开始，则更加偏向经验学科。前三章我们讨论了 ONE 如何想象世界：为什么出发，想成为什么，又怎样落到卡片、排序、已读、发现和 AI 回复上。到了这一章，产品终于要离开会议室、设计稿和发布会，去见真正的人。

用户不会读 PRD，也不会自动继承产品团队的上下文。他们不会因为我们经历了多少争论，就多给一个入口三秒钟；也不会因为某个设计是排除过许多更差方案之后的最优解，就天然理解它、支持它。他们只会在自己的工作里判断：这个东西有没有帮到我，有没有打扰我，有没有让我少做一步，还是让我多背一层负担。

所以这一章不是简单的用户反馈汇总，而是 ONE 带着自己的理想和设计去见用户之后，用户如何验收它、拆穿它，质疑它，允许它——有时候也点亮它的其它方向。

To B or Not To B，that is the question.

2025 年 7 月 3 日，产品师兄拉我一起和顾家家居的两位前线 BD 同学沟通。我们去了 C6 楼下的 M Stand，点了四杯拿铁。两位 BD 聊起顾家当时正在定制开发一个内部的轻量 Teambition ，呈现在 PC 端的主页，解决各种工作信息分散在钉钉各场域的痛点。顾家的痛点和 ONE 的底层技术几乎完全重合——他们也是希望 AI 能够从海量的群聊消息、日程、历史上下文中，自动把待办事项摘取出来，进行汇总和智能推荐。

这本该是一个绝佳的、能验证大模型进入真实业务语境的共创场景。但在沟通现场，当聊到我们当前主要在设计移动端时，对面的两位前线 BD 同学突然沉默了，并且彼此对视了一下，眼神中流露出了明显的犹疑。

回到项目室，师兄私下把我拉到一边谈话，语气非常坦率：「这个后面优先级排低、甚至不考虑了。」

师兄告诉我，大老板的希望肯定是我们 for 所有钉钉用户的，甚至可以当做 「To C 效率工具」来做。这种针对大客户、在 PC 端搞深度定制的业务语境，不是上面想要的宏大故事。虽然现在上面也在讨论收费目标，但走这条路，最终肯定收不成钱。

用户如何买单离场

> 「有钱的捧个钱场，没钱的捧个人场。」

首先，用户表达认可的最直接的方式是花钱。

其次，用户表达认可的方式不只是花钱——有时候用户再认可也不会花钱，这和场景惯性下的心理预期有关。

2025年07月21日 20:44



我在办公IM消息场域愿意为了什么花钱💰：

1、【对标原本要花时间】
(a) [强意愿]沟通 dirty work（如果能做好）（比如：打包消息历史+解释前因后果，自动发送给待沟通方）（会议时间协调也算，但我几乎不愿意花钱）；
(b) [弱意愿]（几乎完全不愿意😌）筛选消息优先级、清红点，etc；
(c) [理想强意愿，甚至终极目标，但当前很难实现]个人代理。

2、【对标原本要花钱】
(a) 实时沟通翻译（语音通话etc）
(b) 回复草稿润色/代写（对标grammarly、写作猫，etc）

3、【对标原本可能有损失】
(a)（已有）消息中识别任务、已读未回etc
(b) 🆕❗离职员工做成一个agent，让其他员工可以咨询。

4、【特殊视觉交互】
视觉动物，没办法（会为了番茄钟特殊主体付费）。

已编辑 >

已读

比如搜索，用户已经习惯了免费的 Google 和百度，而且他们起码达到了「可用」门槛。又比如网游，很难让已经习惯了先免费入场再买装备买道具的用户，先付费买断，即便买断价格可能比后续可能买装备道具的钱便宜很多。

在企业产品中尤甚。付钱的人、使用的人、受影响的人，常常不是同一群人。老板买单，员工使用；管理员开权限，一线员工承受新流程；发信人获得确定性，收信人承担响应压力；采购决策者看组织效率，实际使用者看自己一天能不能少被打扰几次。

所以判断一个用户是否接受产品，不能只听他说「挺好」还是「不好」。更要看他愿意为这个产品付出什么。

他可以付钱，可以付时间，可以付注意力，可以付迁移成本，可以付组织授权，也可以付心理安全感。

对 ONE 来说，用户愿意每天打开，是一种买单；愿意从老钉钉路径迁移到 ONE，是更重的买单；愿意接受系统主动排序、提醒和已读，是心理安全感层面的买单；客户愿意把真实业务流程交给 ONE，则是组织授权层面的买单。

离场也不一定是卸载。企业软件里，用户很多时候没有真正「离开」的自由。他可以不点，不看，不迁移，不推荐，不授权，不把它纳入自己的真实工作流。功能还在页面上，入口还在首页里，后台也许还能看到曝光，但用户实际上已经离开了。

这也是为什么用户研究不能只听嘴上的喜欢。

一个用户说「这个功能不错」，却不愿意多点一次，不愿意改变路径，不愿意让团队使用，也不愿意为它承担一点风险，那么这个「不错」就很轻。相反，一个用户没有说太多好话，却愿意把自己的工作流程交出来，愿意告诉你哪些消息必须被及时响应，愿意围绕这个系统调整自己的操作，这才是更重的信号。

用户最诚实的回答，往往不在嘴上，而在成本里。

ONE 的问题也常常出在这里。我们以为用户在评价「这个产品好不好」，但用户其实在衡量：「我要为它多付出什么？」

我在「设计」章始终没有提 one 中发现的设计，一方面因为我没有参与过发现的设计开发，另一方面因为我自己可能也不是它的用户。

我想象不到，在什么情况下，我需要在工作中打开一个视频流开始刷，也想象不到我是一个老板时，出于什么立场会给员工购买这种服务。

预期与履约

用户可没有参加过 PRD 评审，他对一款产品的预期，来自于产品的名称、交互以及他看到的宣称这个产品能做什么的广告。

ONE 的 slogan 很鼓舞人心：「AI 工作信息流」「专属工作秘书」「让事找人」。不过，宣传越模糊，越漂亮，用户的心理预期也会越高。

如果你说自己是提醒器，用户看到提醒，大概也就接受了；如果你说自己是秘书，用户就会期待你有秘书的分寸和能力，搞不好还会期待一个钢铁侠的小辣椒。

这就是预期与履约之间的距离。

ONE 并非没有价值。它能整理，能排序，能总结，能提醒，也能识别一些待办。但当它被包装成一个 AI 工作入口、一个专属工作秘书时，当整个钉钉赋予它首页左下角这个硕大的 AI 入口时，用户自然会拿更高的尺子来量它。

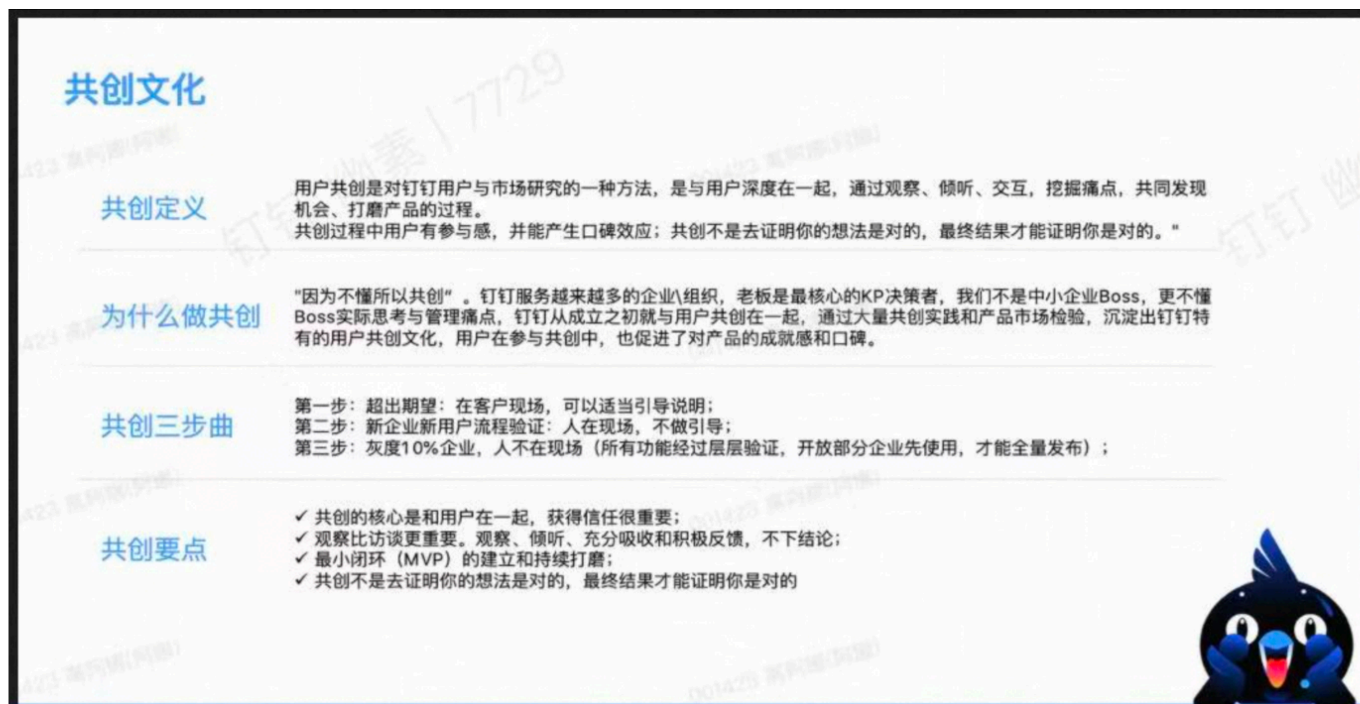
第三章讲过，设计把事情端到了用户面前。用户章要问的是：用户是否认为这算履约。

很多时候，用户并不是拒绝产品说出口的价值，而是在拒绝产品没说出口的代价。

你说帮我整理工作，但没有说会改变我的已读状态；你说让我少漏消息，但没有说会多一个容易误触的入口；你说给我学习内容，但没有说它会在我处理工作时突然出现。

钉钉的共创文化

钉钉并非不重视用户，相反，钉钉有很强的共创文化。「与用户共创」的理念，是我在钉钉蒸馏到的最重要权重之一。要从共创用户——也就是钉钉的客户——身上去发现需求、挖掘需求、做出有用的产品设计。



我很喜欢共创方法论里的一句话：共创不是去证明你的想法是对的，最终结果才能证明你是对的。

这句话说得很好。好的共创，不是把产品拿给用户看，然后等待一句「不错」；也不是找几个客户站台，给发布会增加说服力。好的共创应该让产品团队和用户深度在一起，通过观察、倾听、交互，挖出真实痛点，发现真实机会。

共创真正有价值的时候，往往不是用户夸产品，而是用户把自己的工作结构摊开给你看。

我们有一家智能人事相关的供应商公司，是 ONE 的共创对象。他们给过一个很具体的场景。他们的客服会在那种既有销售、又有客户、又有客服的群里。群里很多消息，是前线销售和客户之间的沟通，客服并不需要每条都看，也不需要每次都出面回复。可是这个群又不能轻易屏蔽，因为一旦客户发消息提问，或者出现需要客服响应的问题，客服必须及时看到并处理。

他们希望系统能做到：只有在客户发消息提问，或者这条消息确实需要客服回复时，才把这个会话的未读红点更明显地标出来，并把该会话置顶前置，成为重要未读。

这个场景非常适合 ONE。

因为 ONE 本质上就在做消息优先级排序。我们规划「重要未读」的原理，本来就要判断三件事：什么样的人，在什么样的群里，发了什么主题的消息。

这不是一个空泛的「帮我看消息」。它有清晰的角色关系，有明确的群场景，有具体的业务责任，也有真实的响应成本。客服不想被所有消息打扰，又不能错过真正需要自己出现的时刻。它要求系统理解的不只是文本，而是工作结构：谁说话，在哪说，说的是什么，这句话是否意味着责任轮到某个人。

这类需求才是共创里最珍贵的东西。用户不是在帮你验证一张卡片，而是在把自己的工作现场拆开给你看。

可惜的是，具体到这件事情，我们始终没有真正做起来。

原因也很现实。它涉及比较长期的开发，需要对群、角色、消息主题、响应责任做持续建设，不是两三天就能出效果的事情。我们一直把它当作一个「重要不紧急」的任务。

而当时 ONE 的开发，被另一套节奏牵着走。

这个节奏我们后来叫「每日一包」。关于它，后面「敏捷」章节会详细讲。简单说，就是每天都要做出让用户有感知的东西，每天都要有东西给无招验收。于是团队很容易被紧急可见的任务拖住，把真正重要但不容易当天展示的事情往后推。

用户给出了真实结构，组织未必接得住。

这件事让我后来很长时间都觉得遗憾。因为它说明，ONE 并不是完全没有遇到正确问题。正确问题出现过，而且很清楚。问题在于，正确问题常常需要时间；而项目当时最稀缺的，正是时间和耐心。

用户研究能指出方向，但开发重心和迭代节奏，决定方向能不能落地。

泥泞的用户现场

有时候共创用户不等于真实用户，共创现场也不等于真实现场。

钉钉共创要求用户有参与感，并能产生口碑效应。这当然有价值，但也意味着用户会深度介入产品过程。介入本身就是沉没成本。尤其当用户也在某种层面上「有求于」钉钉时，他对产品的评判就会变得更委婉、更合作，也更愿意替产品补全意义。

当内测玩家，不是正式服玩家

虽然说，钉钉「共创」的要求是:「不是去证明你的想法是对的，最终结果才能证明你是对的」，但是「共创」过程要求用户有参与感，并能产生「口碑效应」，这也就意味着需要用户的深度参与和介入，而这种介入代表着用户的沉没成本。尤其当用户也在某种层面上「有求于」钉钉的时候，这种介入会让用户对产品做出更委婉的评判。

这让我想到网易互娱的《射雕》。2024 年 3 月公测的时候，这款重金买下金庸射雕三部曲的版权的 MMORPG，雄心壮志，宣称要做「150 年长线运营」。传闻 6 年投入 10 亿， 600 人团队打造。直到 25 年 9 月，项目发布停运公告，短短一年半耳。

上线前，网易内部寄予其厚望，高评分通过多轮内部专家评审，上线前的评级应该是 A 或者 S（对比蛋仔派对是 B）；但实际上线后数据惨淡，迅速凋零。

核心问题是内部评审、内测用户和真实市场之间，发生了断裂。

项目组精心挑了一组内测玩家，内测期间又密集发福利、深度服务和沟通，可以说用一种人弥补上了真实玩家和真实游戏之间的gap，让少数精选玩家理解策划的意图、容忍游戏阶段性的不成熟，甚至参与游戏意义的建构。可是正式上线之后，真实玩家不会自动继承这些上下文。他们没有享受到内测阶段密集的福利，也不会因为设计者曾经在游戏以外的场域，解释过很多次，就天然理解其中的设计。

于是，内测阶段被验证过的「用户喜欢」，到了真实市场里验证为一种梦幻泡影。

ONE 也有一样的问题。

我们的「内测玩家」，也就是共创用户，他们愿意给时间，愿意反馈，愿意理解产品尚未成熟的地方，也更容易被产品团队的叙事、演示和陪伴影响。但共创用户不等于真实大量使用 ONE 的用户：真实用户不会继承共创时的上下文。

共创用户见过产品经理，普通用户只见到入口。

共创用户知道我们想做什么，普通用户只知道这个东西有没有在三秒内帮到他。

共创用户可以被密集运营，真实用户只能被产品本身运营。

比如说 ONE 放在钉钉首页左下角的入口，共创用户知道我们已经排除了不少更加激进和打扰的选项，选择了一个影响相对较温和的；但直接接触上线产品的内外网用户，只会感觉这个入口很容易误触，以往习惯点这里去消息tab，被强行影响了使用习惯。

内测玩家会替产品补全意义，正式用户只验收眼前价值。前者体验的是产品加服务，后者体验的仅仅是产品本身。

这就是为什么很多共创里「还可以」的设计，上线后会显得突然刺眼。共创中的解释、陪伴和耐心，在正式用户那里都消失了。剩下的只有一个入口、一张卡片、一次误触、一个关不掉的模块。

9:54 

日历 待办 3 DING 打卡 通话

消息 置顶 未读

智能招聘助理

一个视频了解钉钉智能招聘

钉钉安全助手

钉钉小秘书

[交互卡片消息]

钉钉管理助手



当用户说谎

做用户研究，最容易犯的错，是把用户说的话当成事实。

这里的「谎言」，不是说用户有意欺骗。更多时候，用户是礼貌的、想象中的、处在组织关系里的。尤其在企业共创里，用户面对的不是一个冷冰冰的问卷，而是一个合作关系、一个供应商、一个平台方、一个可能对自己组织有价值的机会。他说话时，会天然留余地。

《The Mom Test》里有一个很有名的提醒：不要问别人「你觉得我的想法好不好」。大多数人会出于礼貌、鼓励或想象，给你一个好听但无用的答案。真正有效的问题，应该问过去事实、具体行为和真实成本。

"Ounce for ounce, there's no better way to learn what customers want and will buy than this wonderful little book. If you want your new product or new business to succeed, start here."

— **John Mullins**
Author of *The New Business Road Test*



Rob Fitzpatrick

THE MOM TEST

How to talk to customers and learn if your business is a good idea when *everyone is lying to you*

一句「挺好的」，也许只是礼貌；一句「以后会用」，也许只是当场想象；一句「这个方向有价值」，也许只是对方向的认可，和真正迁移工作流之间，还隔着很长一段路。

企业场景里，这层偏差更重。

老板说「这个对员工很好」，员工未必这么感受；管理员说「可以推」，一线未必愿意接；共创客户说「方向不错」，也未必意味着正式上线后，普通用户能忍受误触、打扰和额外路径。

用户还会说一种「想象中的真话」。

比如学习内容。理论上，没有人会反对学习；理论上，paper、行业资讯、管理知识都很有价值。可当用户正在处理消息、日程和待办时，系统突然把他带进一个学习流，实际感受可能就是「像广告」「占地方」「工作时不想看」。

他并没有撒谎。他只是在两个不同场景里，给出了两个都真实的答案。

所以用户的话当然要听，但不能照单全收。要听话里的意思，也要看话后的成本；要看他当场怎么说，也要看他回去怎么做；要看他愿不愿意多点一次，愿不愿意改习惯，愿不愿意让团队使用，愿不愿意把真实流程交出来。

产品研究里最诚实的答案，常常不在嘴上，而在这些成本里。

当设计在计划外的用户群体奏效

ONE 的核心共创对象一直是老板，KP，管理者，信息化负责人和采购决策者。在第二章「定位」的讨论中，也阐释过 ONE 的定位是面向消息爆炸的枢纽角色和管理者，故而也是优先围绕移动端展开设计。

这个判断有道理，管理者的消息确实更多，决策链路更长，也更需要信息过滤。但管理者毕竟凤毛麟角，真实大量使用 ONE 的人，是沉默的大多数。

当时有一个产品同事去广东拜访碧桂园。对方领导层非常重视 ONE 的能力，希望用卡片的形式，动态、智能地安排保安和保洁每天的工作内容和顺序。

比如先巡逻，再清洁，再拍照反馈。做完一个任务，就 check 掉一张卡片。

这个 case 很具体，也很通情达理。基层和流动性比较高的工作者，不需要学习复杂系统，也不需要自己判断优先级，只要按 ONE 的提醒往下执行即可。

他周日赶回来参加了那周的汇报，兴冲冲向无招介绍了这个 case。

不过无招并不欣赏这个共创 case。他的判断是，ONE 不是要服务保安保洁，而是要服务老板、管理者和高净值人群。

其实从这个分歧点延伸出去，不难发现，主动服务里有一个很难绕开的分岔：_**越是主动、越有思考的工作者，越需要保留对自己工作顺序和工作方式的掌控。**_AI 可以辅助他，可以记忆，可以提醒，但不该用很多强规则打断他已经形成的工作节奏。

反而是工作内容相对标准、任务链路相对明确的人，更可能接受卡片式的主动安排。

这种计划外的用户群体比计划内的用户群体更诚实。因为他们不是被产品叙事召唤来的，而是被具体价值吸引来的。一个设计在目标用户那里不成立，却在另一群人那里成立，这不是偏题，可能恰恰是在提醒我们：真正的产品机会不在原先想象的地方。

这就回到了最开始的问题：ONE 到底先服务谁？

如果服务管理者，就不能假装他们愿意被系统安排工作顺序。

如果服务执行者，又不能假装这是一个高净值人群产品。

定性研究：当用户是无招

写到这里，就不得不回到无招。

不是因为用户章应该突然变成一段老板画像，而是因为前面的许多问题走到这里，会集中到一个更深的原因上：ONE 面对的「用户」，并不只有客户、共创对象和普通员工，也包括一个极强势、极高权重、极有产品表达欲的用户——无招本人。

用户研究一般分为定性研究和定量研究。各司各组织的策略倾向不同。就我观察，字节会更喜欢做定量研究，强调北极星指标、漏斗和数据验证；钉钉更喜欢做定性研究，强调共创中的具体用户、具体场景、具体需求。

有的时候我真的不太懂为什么大家特别区分 to B 和 to C 的产品，或者特别区分前台、中台、后台的产品。这些只是标签，是行业为了方便沟通发明的分类框架。但他们的本质是一样的：你要明确你的用户是谁，用户的目标是什么。我们的目标就是帮他达成他的目标。我曾经在网易伏羲 lab 工作。我们的客户是内部的游戏工作室，比如我服务过倩女端游，服务过逆水寒手游，工作室的目标是什么？工作室的目标是玩家的粘性，付费。他们采购伏羲的服务，本质上是希望。外包出去一部分工作目标给我们达成。无论他们提的需求是什么，我们在目标上。要和工作室对齐，这才是真正的以心印心，心心不异。

在钉钉这里，服务无招也是一样的，和服务 C 端群体没有什么很大的区别。只是前者更加定性研究，或者更加定量研究而已。

从这个意义上说，无招既是钉钉的大老板，也是钉钉的大用户。研究和服务无招，也是一种用户共创、定性研究、一种特殊情况的讨论。

但定性研究不是听到一个用户说什么，就原样照做。定性研究要挖原因，挖动机，挖情绪，挖真实痛点，挖底层行为逻辑。它不只看数据多少，也不只问比例，而要搞清楚：用户为什么这么做，怎么想，在意什么，隐藏诉求是什么。

要服务你的服务对象的目标，而不要服务他说出口的需求，而要观察他在现实和社媒上如何自我表达，生活中究竟如何行为。

他的头像是一个武侠形象的黑白背影，上面写着「无招」两个字。这个符号本身就很有意思：武侠、背影、招式之外的招式，既有江湖气，也有强烈的自我命名。再看他过去公开演讲里表达的态度，看他在会议里如何提问、如何打断、如何兴奋、如何不耐烦，看他自己的钉钉使用数据、使用习惯和提出的具体业务需求，都能看见一套相当一致的产品人格。

他是一个高密度、高能动、高控制欲的用户。他处理大量消息、会议、组织信息，需要快速推进，需要确定性，也需要一个系统把事情往前推。

但无招不是普通用户。

他的权力位置、信息密度、注意力阈值、工作节奏，都和普通员工不同。**他是 CEO，是发信人，是组织目标的承压者，也是钉钉历史成功经验的携带者。**他觉得已读重要，普通员工可能感到的是被迫承担；他想要强推进，普通用户需要的是缓冲和控制；他喜欢主动掌控，普通员工可能先问「我能不能关」。

更重要的是，无招回归钉钉时，身上背的不只是一个用户的痛点。

2025 年 3 月底，阿里拟收购无招创业公司「两氢一氧」的投资人股份，无招回任钉钉 CEO。阿里当时的主线是「用户为先、AI 驱动」，钉钉被放在 AI To B 的关键入口位置。到 2025 年底，阿里合伙人体系在 2025 财年从 26 人缩到 17 人，席位更加稀缺。与此同时，周靖人因通义 / Qwen 体系晋升合伙人，吴嘉也被放到千问 C 端事业群，负责千问、夸克、AI 硬件、UC 等入口型业务。

AI 时代的「入场券」正在被重新分配，阿里内部的牌桌也在重新盘点坐席。合伙人身份正在从历史荣誉变成现役战功，从元老共同体变成战略作战席位。

张勇、戴珊、彭蕾、俞永福等人的退出，意味着上一阶段阿里的组织 memory 被压缩；蒋凡进入合伙委员会，则意味着阿里重新把电商主战场交给能打仗的人。吴泳铭时代的阿里，不再试图维持每个山头的体面，而是在「电商」和「AI + 云」两条主线上重排座次。

在这样的背景下，无招回归钉钉，就被寄予别样的厚望。背井离乡的创始人重归故里，他必须证明钉钉不是旧时代的企业 IM 和 OA，而是能成为阿里 AI To B 的关键入口。吴妈召他回宫，恐怕也不是想要 yesterday once more，而是想到 2014 年的无招，破釜沉舟，草创新业，希望这匹老骥伏枥，尚能饭耳。

所以 ONE 的压力，从一开始就不是纯产品压力。它背后有一道更硬的组织命题。

这也解释了为什么 ONE 这个无招回归后最重视的项目，在初始阶段过度追求「大入口」「新形态」「发布会表达」。因为它服务的不只是用户，也服务无招的自我价值证明，以及他在阿里新权力结构里的位置证明。

用户需求是第一层，集团战略是第二层，个人再入场是第三层。ONE 的种种纠结挣扎，恰恰来自这三层目标经常不完全一致。

不过，老板的身份认同，不是用户的目标。

这句话后来反复在我脑子里回响。无招作为用户是真实的，但他是一个极端用户；无招作为 CEO 是关键的，但他不是普通员工；无招作为产品经理是敏锐的，但他的兴奋点不等于大众用户的兴奋点。

如果把一个极端用户的痛点，当成全体用户的路径；把一个 CEO 的身份任务，当成产品本身的用户目标；把一个组织战役的胜负，压到一个移动端入口上，产品就容易走进怪圈。

延伸：「千人千面」的AI产品服务，本质是奢侈服务下放

还有一个后来才慢慢想明白的点：AI 的一大价值，是把奢侈品级别的服务下放给普通人。

奢华酒店会记住客人的偏好：喜欢什么水果、枕头要多高、欢迎卡写什么称呼、上一次住在哪一间房。普通酒店通常做不到。不是不想做，而是成本不允许。一个人能被记住，是因为他足够重要，或者付了足够多的钱。

工作系统里也是一样。老板、核心客户、关键人物，天然有人替他筛信息、排优先级、做提醒、记偏好。他们已经享受了人工个性化服务。普通员工没有。普通用户面对的是同一套默认规则、同一块入口、同一种红点、同一份信息洪水。

所以，越是已经被人肉个性化服务的人，越不容易意识到普通用户为什么需要「自定义自己的 ONE」。无招看到的 ONE，本来就不是一个标准用户看到的 ONE；**他的反馈链路、人工解释、问题响应、产品修补，天然围着他转。对他来说，系统不够个性化，不构成强痛点；对普通用户来说，这恰恰是 AI 最应该补上的差距。**

这也是 ONE 当时很可惜的地方。ONE 早期经常试图用一个个硬规则，去 cater to 以无招为首的少数 KP，然后期待它自然服务好大多数人。

可现实是，大多数人的工作，并不是少数 KP 工作方式的缩小版。用户真正需要的，可能不是又多一个统一入口，而是一个逐渐学会「我是谁」的入口。AI 如果只能把老板级体验再送给老板，价值并不大；它更该做的，

是把过去只有少数人享受得起的贴身服务，变成每个普通工作者也能拥有的日常。

谁重要、哪个群重要、什么消息该回、什么时候该提醒、发现要不要看、入口放哪里舒服——这些事情都高度依赖个人角色、组织关系、工作阶段和心理阈值。一个老板、一个客服、一个销售、一个普通员工、一个信息化负责人、一个项目经理，对「重要」的理解完全不同。

到最后，我们再回看各种用户反馈，会发现一个有意思的现象：

除了明确提出想关掉「发现」入口，以及解决左下角 AI 入口误触的问题，用户需求分布非常不集中。

这个现象不能简单理解成「用户没有需求」，也不能理解成「用户说不清楚」——它更可能说明：用户真正需要的，不是又一个统一硬规则，而是更**个性化的服务**。

于是又可以 call back 设计章的尾声：形式不该凌驾于内容。

以有涯的设计求无涯的用户需求，无异舍本逐末，缘木求鱼。

敏捷第五

钉钉是个勤行。

什么是勤行？勤行就是早餐店，杂耍摊，手工业，是冬练三九，夏练三伏。

这句话没有一丝讽刺。恰恰相反，这是我对钉钉产研最深的敬意和赞美。我见过不少团队，也在不同公司做过产品，但钉钉的响应速度和开发配合程度，确实罕见。

有个工龄六年多的同事后来去了字节，某个周末聊起来，他感慨：「我感觉字节没钉钉快。」我问是哪方面的快，他秒答：「开发速度。」

「字节没有钉钉快？」

「没有钉钉快。」

半晌他又补了一句：「字节流程还是慢。」

于是我们一起叹气：「真是白瞎了。」

听到「开发速度」「流程」，我一下子想起项目室的日日夜夜。

老板们上午在群里提的要求，晚上必须能打进彩虹包里验收。今天改交互，今天出稿，今天调整规则，今天进包，今天指出问题，今天给方案。很多时候，连「明天」都显得奢侈。

我们称这套机制为「每日一包」。

这套机制当然有它的好处。它让团队一直在动，让问题不过夜，让产品每天都有新东西。对于一个新项目，尤其在发布会前后，这种速度会带来强烈的战时感：所有人都在场，所有问题都被盯着，所有变化都能迅速看见。

ONE 能在短时间内形成复杂的移动端框架，能把消息、待办、日程、会议、发现、AI 指令、播报、总结等一系列能力塞进一个新入口，靠的正是这套勤行手艺。

所以，复盘 ONE 时，不能把问题归结为「团队不努力」。

甚至恰恰相反，很多问题发生在一种过度努力之中。

在「发心第一」章节中我提到，「智能是平权的，但是 context 是不平权的」。其实还有一件东西是不平权的，那就是人的创造性和主观能动性。

针对「搭建企业Agent」的愿景和效果，其实是非常共识的。大家对于 AI 提效的愿景已经非常趋同一致，拉任何一个人上台，照「I have a dream」的发言稿起手，也能说出不少子丑寅卯来。

说起来原理也简单。那些你放心布置给一个刚入职不久的同事干的活，将来都可以放心交给 AI：查资料、整理会议、建日程、写周报、跟进事项、汇总进展、把散在各处的信息归拢成一个可行动的判断。它不用睡觉，不怕重复，理论上还能「越用越懂你」。

然而，知道最终幻想是什么样，实际情况与之仍然隔着一片 limbo。

造成实现效果差异的，除了各产品既有的数据和生态护城河，最大的变量就是产研团队本身。一个团队是否能把愿景切成路径，把路径切成任务，把任务做成可用产品；是否能在每天都有人催促变化时，仍然保留一点建设地基的耐心；是否能在「看起来很快」和「真的在接近问题」之间，分出轻重。

这就是「敏捷第五」要写的事。

上一章写用户。用户告诉我们，ONE 里有些东西成立，有些东西不成立；有些行为用户愿意迁移，有些行为用户不愿意交出来。到了这一章，问题往下走一层：既然真实问题已经出现，组织有没有能力持续做对的事？

这不是简单的开发速度问题。

钉钉很快。ONE 也很快。问题在于，跑得很快以后，谁来决定方向；每天都有交付以后，什么会被做深，什么会被挤掉；敏捷究竟是在接近真实问题，还是在更快地、被动地响应，一条又一条新的信息流。

迭代与汇报节奏

我有朋友在腾讯 WXG 工作，有时候私下里我们会对比两边产研的节奏，发现一个很有意思的现象：微信是默认一动不如一静，如无必要，不增加产品改动；钉钉反之。

（叠甲：到这一步，我并没有认为二者孰优孰劣，这只是由产品性质和风格决定的、中性的产品迭代的选择。有时候用户也乐意看到产研团队折腾，毕竟钉钉是付费的 SaaS 产品，有时候你不折腾，用户觉得钱没花到位，或者产研团队在尸位素餐。）

钉钉的敏捷，首先体现在对无招信息流的响应。

一般早上九点到十一点，有问题，无招会直接发给技术负责人或管理层群。管理层各自响应，领走问题。一个小时内反馈打算怎么处理，二十四小时内交付。

这很像一套最高优先级的问题单系统。

AI 创建日历不符合预期，立刻查；消息列表抖动，立刻看；导航栏里标题截断，立刻讨论；用户说「不智能」，立刻追问原因；某个外部截图、某个内网帖子、某个客户反馈、某个竞品动作，只要进入无招视野，就会获得极高权重。

这套机制的好处显而易见。老板不隔层，不装聋，不等周报。看到问题就问，问了就要解。许多企业做不到这一点。很多问题在层层传递中已经失真，到了老板那里只剩一句「持续优化中」。钉钉不是。钉钉很多时候是问题刚冒头，刀已经递过来了。

但这套机制也有另一面。

敏捷从来不是抽象的。组织总是对某些人、某些事、某些信号更敏捷。问题并不是「钉钉敏不敏捷」，而是「钉钉对谁敏捷」。

当无招的信息流成为最高优先级队列，团队就会天然围着它转。被他看见的问题，会迅速进入战时处理；没有被他看见的问题，即使用户反复提、产品反复想，也容易留在「重要不紧急」。

这就是 ONE 后期很典型的处境。

无招看到列表抖动，列表抖动要解。无招觉得消息助理不智能，消息助理要马上给出体感。无招对 Agent OS 有新想法，框架立刻进入讨论。可用户个性化、自定义主页、偏好学习、长期记忆、重要消息反馈闭环，这些更像地基的事情，很难用一天的包证明自己。

于是，敏捷像一把很快的刀。

它能切开眼前的结，也能把长期问题切碎，分散到每天的改动里。

每日一包

每日一包，本质上是一套可见变化的生产机制。

我在刚接触到这个概念的时候，说实话我觉得它很好，是一种对用户的负责和一种对产研团队自己的严格要求。但在具体的执行中，我越来越不敢完全苟同，并且意识到其中的风险和害处。

它退化成了高管意志下的一场实验。

它偏爱：今天能看见的，今天能截图的，今天能被老板验收的，今天能写进 changelog 的。

它不喜欢：需要长期建模的，需要打通底层数据的，需要反复验证才有效的，一开始看不出效果、但半年后决定上限的。

这不是某个人的错，是机制本身的偏好。

比如，我一直想做用户个性化设计自己的 ONE 主页。每个人的工作不一样，重要的人不一样，常用模块不一样，对提醒的忍耐度也不一样。一个销售、一个客服、一个管理者、一个普通研发、一个信息化负责人，理想中的 ONE 不可能长成同一张脸。

这件事在用户章里已经有答案：用户需要的，未必是又一个统一硬规则，而是系统逐渐学会「对我来说，什么重要」。

可这件事很难进入每日一包。

它要做偏好，要做配置，要做记忆，要做反馈闭环，要允许用户关掉一些东西、固定一些东西、调整一些东西。它不会在第一天就让人惊呼「哇，变了」。甚至它最好的状态，是用户用久了之后觉得顺，觉得少被打扰，觉得这个系统越来越像自己的。

这种价值很难汇报。

更关键的是，无招并没有这个强诉求。

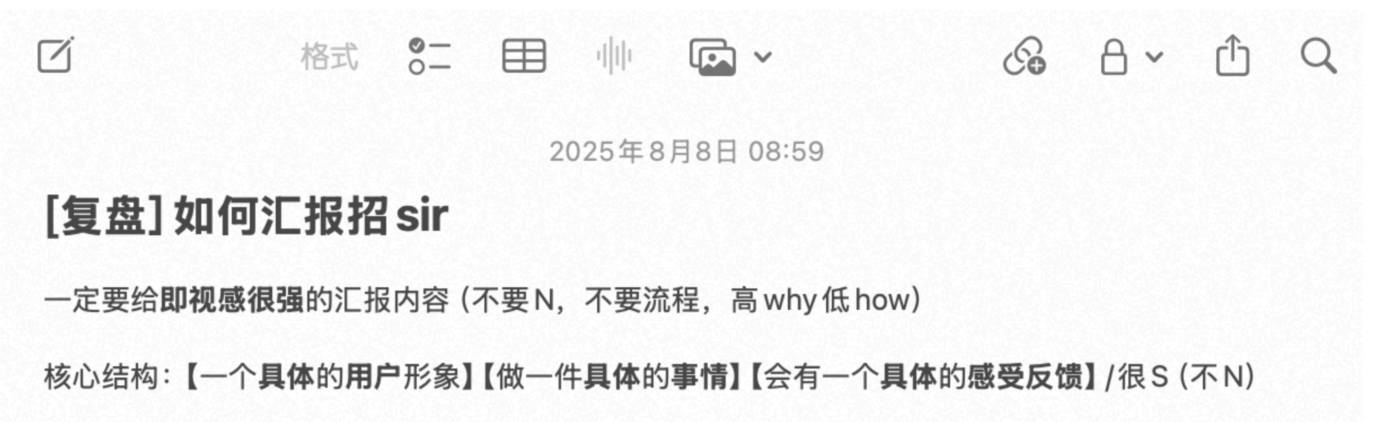
因为他看到的产品，本来就不是标准用户看到的产品。围绕他的反馈链路、问题响应、人工解释、产研支持，已经构成了一套「人工个性化」。他不需要自己慢慢配置系统，也不需要忍受一个陌生入口长期不懂自己。

他的问题被看见得太快了，他的需求被响应得太快了。

于是，正确问题被留在旁边。每天交付的，是更容易被看见的变化。

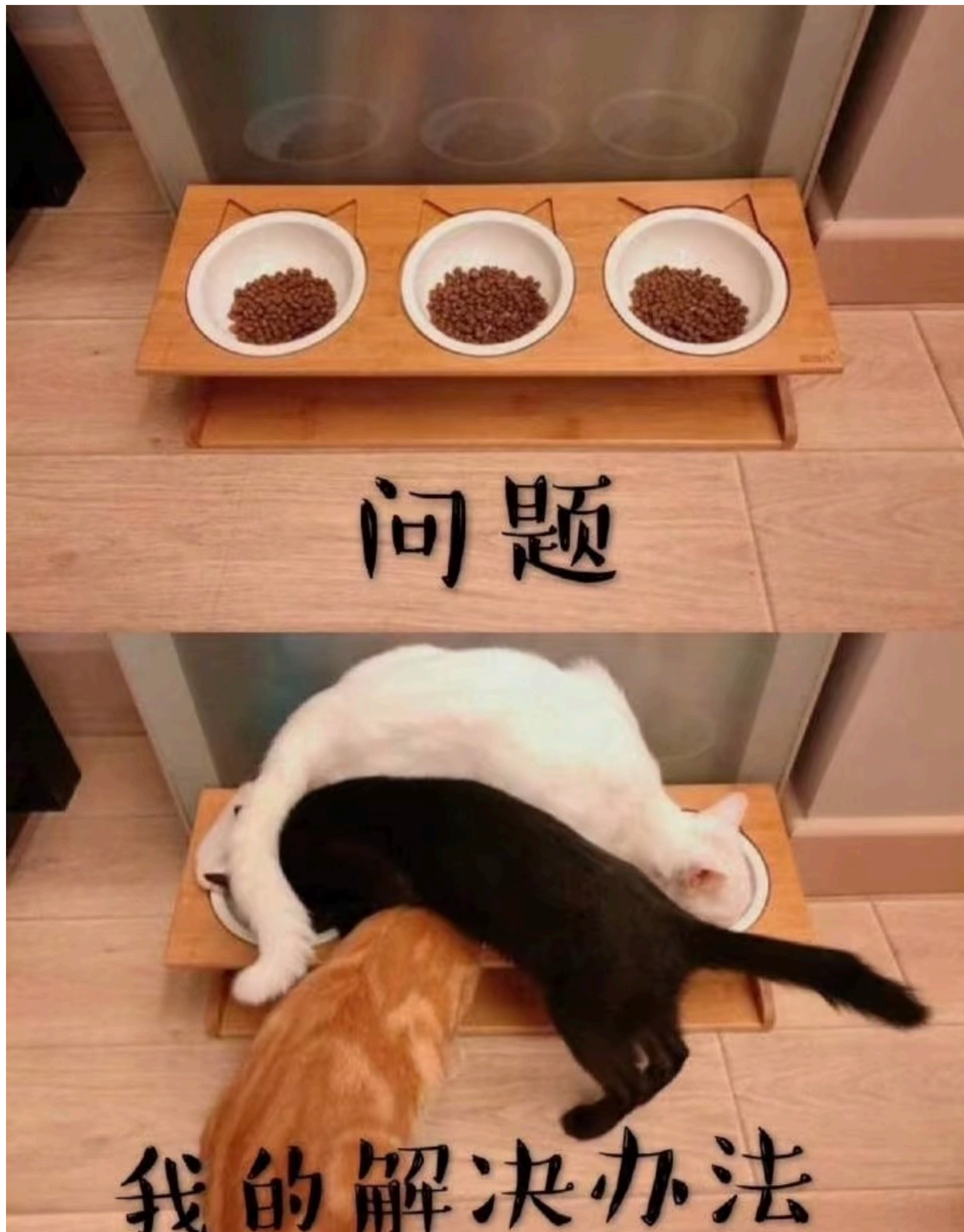
汇报技巧

翻到去年 8 月份记录的无招汇报技巧，截图留念，以后有需要的朋友自取。



Beating Around the Bush

其实在产品讨论和开发阶段，我就感到我们「在回避真正问题、只做表面优化」。



这个底层是非常傲慢的，我们希望一下子上来就猜准用户想要什么，然后推荐。有时候客户明明明确说了，有一些个性化的需求，希望能指挥AI做xxx，但我们纠结在自己的能力域里，鼓励用户削足适履。明明调研下来，绝大多数用户都不能在做好心理准备之前，直接点开消息看未读，我们还硬着头皮做。

用户怕下一张卡片是什么，于是做横滑头像，让他先看到一点预告；用户怕完全划过去就已读，于是做 Peekaboo，让他在没有完全松手前处于一种「预读」；用户觉得消息不清楚，于是补摘要、调排序、改展

现；用户对卡片焦虑，就想办法给更多跳过、展开、预览。

这些工作不能说没有价值。每一项都是在缓解痛感。

但它们共同绕开了同一个根问题：这个已读到底该不该发生？

响应快，是不断回答「怎样让这个设计没那么痛」。迭代深，是敢问「这个设计是不是一开始就不该这样」。

ONE 在已读问题上，大量时间花在前者。我们很敏捷地修补了很多地方，却始终没有敏捷地回到最初的判断。这就像房梁歪了，却每天换窗帘、擦地板、调灯光。屋子确实越来越像样，住在里面的人还是觉得不安。

一鼓作气，再而衰，三而竭

士气要靠胜仗养。

「夫战，勇气也。一鼓作气，再而衰，三而竭。」这句话放在项目里也一样。

团队可以加班，可以赶发布会，可以日更，可以随叫随到。人不怕累到极点，人怕累了很久，却不知道自己赢在哪里。

胜仗可以有很多种。用户真的用起来，是胜仗；数据长出来，是胜仗；一个难题被解决，是胜仗；团队发现自己比昨天更接近正确方向，也是胜仗。产品人和工程师都不怕辛苦，只要辛苦之后能看见东西。

可如果每天都在改表面、追热点、救火、补洞、改口径、赶汇报，人的意义感会被慢慢掏空。

ONE 到后期，团队人心的问题已经开始出现。这个问题很复杂，不能只算在某个人头上，也不能只算在某个产品决策头上。它和组织的目标、节奏、信任、归属、话语权都有关系。这里暂且按下，到下一章「秩序第六」再写。

这里只提一句：没有胜仗的敏捷，很快就会变成消耗。

我记得很清楚的是，去年11 月 14 日早上很早，我和另一个产品很早去找无招汇报。才 8 点多，~~HR~~ 秘书也在会议室，在砧板上切新鲜水果榨汁。那天本来汇报一个改动可能比较大的新方案，有些心焦，~~HR~~ 秘书的在场给我的存在感就格外强烈，她一刀一刀不紧不慢的声音，就像直接切在人的耳膜上。

汇报到一半，~~HR~~ 秘书把果汁递给无招，后者突然问我们，「xx（另一个产品），幽素，你知道你们最大的问题是什么吗？」

我们相顾无言。

他说，「你们最大的失败是没有团结好团队。人心散了，你们知道吗？」

我哑口无言。

这是我的，我们的失败吗？

> 非常抱歉地更正：

> 当时切水果的同事，和同在现场的同事确认后，系秘书而非 HR，原文也不是指我现在的 HR。我现在的 HR 人非常好。

90 分 Agent：高分小题 v.s. 真实世界

后来内部开始讲「90 分 Agent」。

这个说法本身没有错。Agent 不能只在 demo 里聪明，也要在真实业务里交付一个用户愿意承认的结果。企业 AI 尤其需要稳定、可控、可评测。

2025 年以后，外部环境也在往这个方向走。最近参加 AI 闭门会的人提到一个判断：大模型公司的商业化重心，可能正在重新回到企业场景。OpenAI、Anthropic 开始做更深的企业服务和工程交付，不再只是卖 API，而是试图通过服务，把模型直接嵌进企业核心业务流程。国内也有阶跃星辰和金蝶合作，企业软件公司和大模型公司开始深度绑定，做下一代企业服务 Agent。

这个趋势很值得注意。它说明 AI 真正进入企业，不会只靠一个聊天框，也不会只靠模型调用。它要进流程，要进系统，要进那些繁琐、具体、反复发生的工作。比如评论区有人说：「什么时候能用上 AI 报销，真不想再贴发票了。」这句话很朴素，却比很多宏大叙事都准确。企业 Agent 的价值，常常就在这种地方：不是炫技，而是把一个人反复厌烦的流程做掉。

所以，90 分 Agent 这个方向当然重要。

问题在于，90 分究竟评的是什么，以及，是不是要为了短期快速达到这个 90 分，对产品削足适履。如果评的是短期交付效果，团队很容易自然地收窄场景、减少变量、提高命中率。比如「已读未回」为了提升成功率，最后限制为只监听直属上级消息。直属上级当然重要，关系也更容易定义，规则更清楚，模型更容易判断，指标更容易好看。

可用户价值也随之变窄。

一个人一个月究竟会忘记几次不回老板消息？他真正容易忘的，可能是客户群里没人接的一句话、同事随口托付的小事、会议后没有落下来的后续动作。这些事情更散、更脏、更难归因，却也更接近真实工作。

没有限制仅上级时，用户说「可回可不回」「有的已经线下沟通过了」；限制到直属上级后，又总是触发不了。前者太脏，后者太窄。为了让功能更像 90 分，系统选择了最容易得分的一类关系。题答对了，场景却变小了。

这就是削足适履最隐蔽的地方。它看起来是在追求交付，实际上是在把用户的脚修成鞋的形状。

90 分 Agent 应该追求稳定交付，但不能为了稳定，把真实世界改造成适合考试的题目。企业 AI 最难的地方，恰恰在于真实世界不干净：信息散在不同系统里，责任藏在人际关系里，流程有例外，权限有边界，用户还有情绪。

一个高分小题，未必是一个好产品。

AI 跃进下的技术债

2025 年，AI 行业最不缺的就是快。

模型快，融资快，demo 快，发布会快，老板转发外部文章也快。今天 Slack 讲 agentic OS，明天 Salesforce 讲 Agentforce，后天某个创业公司又发一条视频，说十个人的活现在一个 Agent 就能做。所有人都在提醒所有人：再不快，就要被时代甩下车。

这种快当然有道理。AI 的窗口期确实短，企业协作产品也确实站在牌桌上。钉钉和 Slack 一样，有消息，有组织关系，有工作流，有企业数据，有天然的上下文。别人开始讲工作场景里的 Agent OS，我们不可能毫无反应。

可是行业叙事只能证明方向值得做，不能替代落地顺序。

要做 Agent OS，先要有能独立站住的 Agent；要做工作入口，先要稳定处理一类工作；要让 AI 主动服务，先要知道它凭什么判断、凭什么执行、错了以后谁来兜底。否则很容易先得到一个像未来的框架，再回头发现里面没有足够结实的骨头。

后来我读 fast.ai 一篇叫《Build to Last》的文章，反而对这件事有了更深的感触。

那篇文章整理了 Jeremy Howard 和 Chris Lattner 的一次长谈。Chris Lattner 是 LLVM、Swift、Mojo 背后很关键的人。文章里有一个很朴素的判断：真正能留下来的软件，靠的不是短期堆砌，而是第一性原理、架构质量和手艺。LLVM 能支撑二十多年，不是因为它追过每一个风口，而是因为它一开始就从根本问题出发，搭了一套能让后来许多语言和系统继续生长的地基。

这话放在 ONE 身上，实在是刺耳的良言。

我们当然要快。AI 时代不快，几乎等于放弃。可是快也分两种。一种快，是因为地基打得好，抽象做得好，团队对系统理解得深，所以越做越顺；另一种快，是今天先补一个口子，明天再糊一个入口，后天再把老板刚提的问题塞进包里。前一种快，是能力；后一种快，是欠债。

Chris Lattner 在那篇文章里还批评了当下流行的 「vibe coding」。他并不是反对用 AI。相反，他承认 AI 有用，也能提高效率。可他反感那种不求甚解的做法：让 Agent 一轮一轮生成代码，像拉老虎机一样，试到某个结果能跑，就以为问题解决了。真正危险的地方，不在于 AI 写了多少代码，而在于人放弃了理解和掌控。

先搭一个未来感很强的入口，先铺一组卡片，先放一个会说话的输入框，先让它看起来像一个 Agent；然后根据老板、客户、竞品、发布会，一轮一轮补洞。每一步都有理由，每一步都能解释，每一步也许都能在当天进包。可是如果团队没有真正理解用户场景、系统边界和长期架构，补丁越多，债越厚。

ONE 的许多问题，都属于这种债。

要做个性化，就要有记忆、偏好和反馈闭环。

要做重要未读，就要理解角色、群关系、消息语义和责任归属。

要做 Agent 间协作，就要让机器人之间能互相感知消息。

要做跨系统执行，就要有权限、审计、失败恢复和长程状态管理。

要做一个真正的 AI 工作入口，就要让每一次行动都有上下文，也有后路。

这些东西都不适合「每日一包」。

它们不容易截图，不容易当天验收，不容易让人晚上看了觉得「又进了一步」。可它们决定了一个 AI 工作产品能不能真正走远。

Chris 还有一句话，大意是：他不喜欢那些做决定时仿佛 AGI 或 ASI 明天就会出现的人；那种疑神疑鬼、焦虑不安、害怕好好生活、害怕建设更美好世界的状态，在他看来很愚蠢，也不实际。

这句话我读到时，心里一沉。

AI 的进步是真的，焦虑也是真的。但如果因为相信明天会有更大的模型，今天就不打地基；因为觉得后天 Agent 会自动解决一切，今天就不认真理解用户和系统；因为外面都在讲未来，里面就急着把一个半成品包装成未来感——这不是拥抱未来，是拿未来替现在逃债。

技术债不只是代码债。在 AI 产品里，技术债还包括：没有理解用户关系，却让模型判断重要性；没有建立反馈闭环，却要求系统越来越懂用户；没有打通权限和审计，却想跨系统执行；没有 dogfooding 到足够痛，却急

着对外说它能成为工作入口。

这些债一开始都很安静。它们不会像 bug 一样立刻报错。它们只是让产品每次多绕一点，每次多补一点，每次都离稳定能力差一点。直到某一天，用户说「不智能」，团队说「我们明明做了很多」；用户说「不敢用」，团队说「我们已经加了预览」；客户说「我想让 AI 按我的业务处理」，团队说「这个暂时重要不紧急」。

然后，债就到期了。

文牍先行

项目越高压，汇报系统越早成熟。

ONE 后期有很多模板。需求列表、每日任务、changelog、周报、问题清单、竞品走查、客户反馈、截图证据。项目还在摇摆，模板已经稳定；能力还没闭环，changelog 已经齐全；用户价值还没沉淀，汇报链路已经很顺。

这也不是纯粹的形式主义。

汇报有它的功能。它稳住老板预期，稳住管理层判断，稳住团队对「事情在推进」的体感。尤其在高压项目里，如果没有文档和汇报，所有人都会更慌。模板像一排临时脚手架，撑住一栋还没完全成形的楼。

但脚手架搭久了，人也会误以为楼已经建好了。

竞品走查、问题清单、截图对比，本来都是好东西。可在高压环境里，它们很容易把产品变成「不要落后」的清单。外部有，我们要有；老板问，我们要答；用户骂，我们要记；本周改了什么，下周要能列出来。

于是团队开始为可汇报生产，而不是为可沉淀生产。

可汇报的东西，常常长得很像进展：多了入口，改了样式，补了规则，修了文案，上了一个新卡片。可沉淀的东西未必热闹：一个更稳的排序反馈闭环，一个更准确的关系理解，一个更可控的已读策略，一个更懂用户的偏好系统。

前者更像每日一包，后者更像长期修渠。

文牍先行，最怕的不是写文档，而是文档比产品更像产品。

小结：拥抱敏捷，警惕敏捷

后来我逐渐意识到，「每日一包」不是一个单纯的研发节奏，而是一套组织记账法。

能在当天进包的，算进展；能截图汇报的，算进展；能让老板晚上验收的，算进展；能写进 changelog 的，也算进展。它们都真实存在，都消耗了人的时间和力气，也确实让产品每天变化。

可是另一类东西很难入账。

用户个性化主页没有入账，消息排序的反馈闭环没有入账，已读状态的根问题没有入账，Agent 之间的消息感知没有入账，长期记忆、权限、审计、失败恢复也没有入账。它们不是不重要，只是不容易在当天显影；不是没人知道，只是不适合在晚上拿出来验收。

于是项目看上去一直在跑，账面上每天都有收入，底下却不断形成隐债。

敏捷若服务于学习，会让产品更快接近真实问题；敏捷若服务于证明，会让项目更快产生变化，也更难承认方向需要重来。

ONE 的许多问题，正在这个缝隙里发生。

它也太勤快了。

它的问题常常不在于没人响应，而在于响应得太快，快到来不及分辨这个问题来自哪里、指向哪里、解决后是否真的让产品更接近用户。

这也是我后来对「敏捷」这个词最深的警惕。

真正的敏捷，不是每天都有东西改，不是老板一问，团队就答，不是一个问题二十四小时内必须交付，也不是把所有长期问题拆成短期补丁。

真正的敏捷，是更快学习，更快校准，更快承认误差，更快把资源转向正确问题。

如果一个团队每天都在动，却没有更接近正确问题，那不叫敏捷，叫奔波。

ONE 给我的教训正在这里。用户已经指过一些路，产品也看见过一些光。可在每日一包、即时响应、外部叙事、汇报链路和组织压力之间，很多正确问题没有被持续做深。它们被记下，被讨论，被放进表格，被标成「重要不紧急」，然后被下一天的紧急问题盖过去。

于是，敏捷走到尽头，就不再只是开发问题。

它开始变成秩序问题：谁定义优先级，谁制造紧迫感，谁拥有最终解释权，谁承担消耗，谁能获得胜仗，谁只是被不断投入火里的柴。

这就到了下一章，「秩序第六」。

秩序第六

> 东方未明，颠倒衣裳。颠之倒之，自公召之。

> 东方未晞，颠倒裳衣。倒之颠之，自公令之。

> 折柳樊圃，狂夫瞿瞿。不能辰夜，不夙则莫。

> ——《诗经・齐风・东方未明》

钉钉搬去阿里未科中心之前，工区在西溪 C 区 C6 楼。C 区的楼多是 U 型结构，中间是一方开阔空廊，四面是层层叠叠的办公区、玻璃、栏杆和人影。站在下面抬头看，会有一种在鸟笼里往外看的感觉。

这和钉三多的雨燕意象正好相反。雨燕可以连续飞行十个月不落地，是速度、耐力和自由的象征；C6 像一只透明鸟笼，光线很好，秩序很好，视野很好，只是天空被折叠成了中庭。

我刚来钉钉，就进了一个秘密项目室，甚至没有一个可注册的正式工位。项目室在 C6 四楼中庭，一个大房间里有四十多个位置，产品、开发、设计都坐在一起，晨会、晚会也都在这里开。四五排桌椅排得很密，空气不大流通，一有人生病，就容易传染周围一片。

如果你喜欢福柯，钉钉无疑是一个格物致知的好地方。《规训与惩罚》里的全景监狱，最要紧的并非「有人正在看你」，而是你不知道自己是否正在被看，于是开始主动把自己训练成适合被看见的人。

C6 的空间结构、钉钉的产品哲学和组织秩序，在这里形成了一种奇妙互文：上级可见、已读未读、DING、群响应、验收包、Scrum 评分、VOC 处理、算粒申请。它们都在处理同一个问题：可见性和确定性。

问题不在于组织不需要可见性。企业协同当然需要确定性。问题在于，当可见性成为最高价值，人就会自然地把时间投向「如何被看见」，而不是「如何真正创造价值」。

飞得最快的鸟，落进一顶透明鸟笼，构成了笼中鸟的隐喻和景观。

什么是钉钉的优先级

上一章提到，钉钉的敏捷首先体现在对无招的需求的最敏捷响应。 我在 C6 楼印象最深、待得最久的地方，除了四楼中庭的封闭项目室，就是五楼东南角无招的会议办公室，以及门口围出来的一小片沙发等待区。

ONE 在开发最紧张的时候，每周至少要做两次汇报（当然，招 sir 有时候也会闪击项目室）。每次在门外会等待，短则十几分钟，半个小时，长则一两个小时的也有，有时候下午 4 点到，慢慢等到快 6 点，又改成晚上 8 点，又改成晚上 10 点，最后变成 12 点以后直接在项目室汇报的也有。

时间在哪，爱就在哪

有一种说法是：「钱花在哪，时间花在哪，爱就在哪。」

同样有说法是：「不要看他说什么，而要看他做什么。」

同理，观察一个企业的优先级怎么安排，不要看它宣称什么，而要看它怎么分配员工的时间。

项目室里有早会，也有晚会。Scrum 要拆每天的任务，SM 要写每周、每天的规划，组织晨会晚会，还要在每周给人打分。无招通常在早上九点到十一点之间，把看到的问题发到技术负责人或管理层群里。管理层各自认领，通常一小时内反馈打算怎么处理，二十四小时内交付。

下班前，要把需求列表和每日任务拆完；晚上，新的包还要继续验收。很多问题不是「今天要不要处理」，而是「今天几点处理完」。

这套节奏塑造了一种很强的响应能力。问题不过夜，反馈不悬空，谁负责，什么时候给结论，几乎都要落在明面上。它确实让组织变快，也让事情不容易烂在半路。

但另一面是，人的时间被切碎了。团队会越来越擅长追光，越来越不擅长自己看路。

你很难保留一整段时间去想一个难问题。你刚想往下挖，群里有新问题；你刚准备推一个长期方向，晚上要进包；你刚想补足上下文，另一个更紧急的事项又来了。

人会越来越擅长立即响应，越来越不擅长完整思考。时间不再是一块可以经营的自留田，而像一个随时会被征走的仓库。谁手握兵符，谁就能调用填充整个仓库；反过来说谁离中心更近，谁的时间就更容易被打断。

错误的注意力导致错花的时间

在钉钉，老板有非常多的焦虑去盯员工工时的付出。谁国庆节没来，谁年初四没有回来加班，谁周日汇报的时候不在。

其实就按人力量来说，钉钉明明是和 Claude Code 的员工可能付出了10倍、20倍、30倍、甚至百倍不止的时间和人力投入，钉钉员工也非常的努力，配合度非常高。但是实际执行的效果就是非常的糟糕，因为时间被错用在了非常不对的地方。

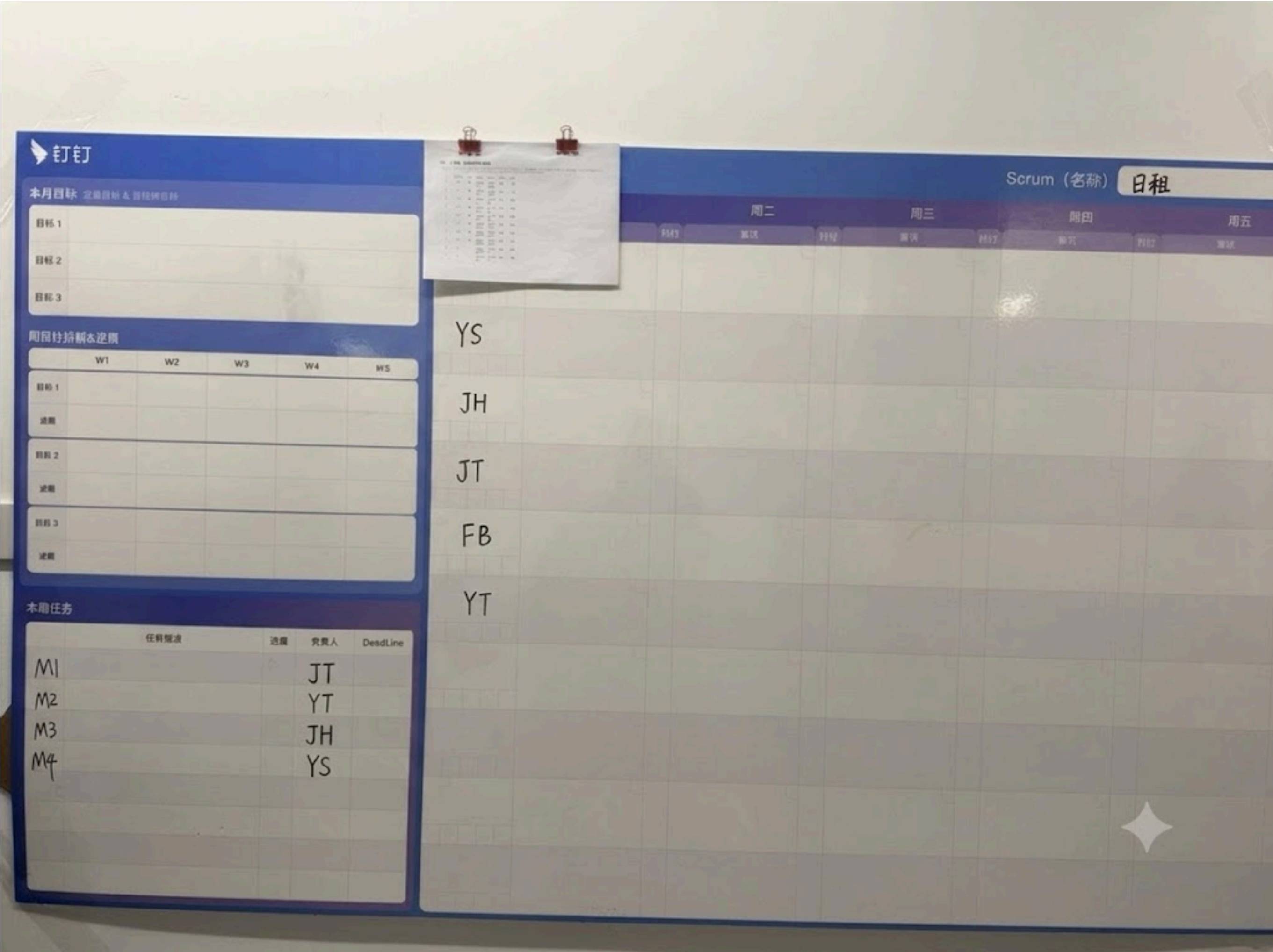
以 skills 为例，老板就想做 skills，并且分配了许多员工去做 OPT（One-Person Team. Having no idea why it has to be 'Team' instead of 'Company'，一开始还以为是美签），但很多时候根本不是 skills 的问题。参见Claude Code，反而在缩减自己 skills 的规模。

基于 Scrum 的陟罚臧否

什么是Scrum

组织真正奖励什么，不看宣传什么，看打分给绩效。

钉钉会分 Scrum。每个 Scrum 有 SM，也就是 Scrum Master，负责写每周、每天的规划，组织早晚会。钉钉有一套早晚会制度，早上九点，晚上是十点或十点半，需围绕着贴在墙上的看板Scrum 里的人一个个对照说明自己做了什么，打算做什么，可能存在什么卡点。



SM 也负责每周给人打等级分。一般普通是 B，好是 B+，不好是 B-，严重不好会打 C。

站在 SM 的角度，这件事也很为难。因为组织要求必须拉开差距，每周总要有人背低分。低分有时是评价，有时也像一种组织配额。总得有人落在后面，队伍才显得被管理过。

有一件发生在个人身上的事情，让我对这个制度至今有所存疑。

清明节前，我请了半天假；节后，又请了一天假。隔着一个周末，前一周和后一周，我连续被打了 B-。

这是我除了春节后请三天假以外，第一次被打 B-。我很困惑，去找 SM 问具体原因，他告诉我就是因为请假。我反复确认，就是因为请假吗？不是我有什么工作做得不到位？他说，你别多想，就是因为请假。

可能有些小小讽刺的是，这个休假的一天半，是我之前整个三月份周末无休换来的调休假。

当组织把可见投入当成默认美德，休息就容易被解释成不投入。哪怕请假是制度允许的权利，哪怕那是我周末加班 8 天（如果钉钉周日工作不算加班的话，也有 4 天）兑换成的 1.5 天调休，哪怕人需要恢复，哪怕工作本身的进度根本不会因此坍塌毁灭，只要评价体系关注「在场」和「待机时长」，正当休息也会蒙上道德阴影。

制度只顾眼前，就会奖励员工学会眼前最容易看见的东西：在线、随时响应、接受传召。健康的制度应该保护长期能力，保护人的恢复，保护真正的创造性产出。可在强响应秩序里，短期可见往往压过长期健康。

于是，人会学得很快，学会在群里出现，学会及时回应，学会把工作切成容易被看见的小块，学会让自己的忙碌留下痕迹。至于那些真正有价值但不显眼的部分，反而要不断自证。

一个评分制度，不只是评价人，也是在对人反向做强化学习训练。

短期和长期臧否

有一个 25 年校招进来的一个设计，她是那种认真的，精益求精的，温和的，爱笑的，总是换位思考的，即便不赞同也会遵守规则的小女孩。她进钉钉之后做了非常多杂事，前段时间被指去做 opt 写一人公司 skill 了。

我私下问她，这真的是你要的吗？如果你将来想要在设计道路上深耕，做这个是不是走进岔路了？她说，一开始也有点难受，但觉得同事们人都很好，服务客户、写 skill、用 AI 做设计，也很有意思，她想学习这块内容。

钉钉 5 月底刚打了最近一次绩效，她的绩效不怎么好看。

何者？

短期看，人帮助团队，做和自己在组织里会如何被评价完全不相关的工作，短期的绩效可能不错。长期看，如果我是 leader，我也很难给苦劳而不是功劳，打高绩效。

另外，钉钉打绩效的方式也比较独属于钉钉，主要是 SM 打。不难想象，如果 SM 的工种和你不同，就容易比较尴尬。一方面，一个跨领域的——比如技术的负责人，对产品设计究竟做得怎么样，除了客观数据和工时之外，相比同工种的负责人，是比较难以有一个足够置信的评价的，不然面试也不会把这个环节叫做「交叉面」了。另一方面，技术主要对技术实现负责，如果他手上有一些好绩效的指标，也更愿意用来管理手下的技术同学。

上有政策，下有对策

可能没有人真正发自内心的喜欢和需求早晚会制度。如果工作量真的反刍饱和，可能倒也还好，开会就当顺带了；对于没那么多工作要做却不得不表演「保持在场」的人来说，就会比较痛苦。

不过，上有政策下有对策。比如说，就有同事建议我晚上 6 点钟直接回家休息，9 点半再来，开完晚会再回家。

设置违反人性的制度，只会促进员工的注意力放在那些制度制定者初衷更不希望看到的地方。

AI带来的管理变革

失落的中层

无招在 2026 年《中国企业家》商界木兰年会上指出：「AI不是让员工更高效的工具，而是彻底重构组织、企业甚至个体的生产关系、决策体系和协同方式的催化剂」同时提出「未来公司没有中层」的判断。核心逻辑是：

1. AI 可以充当"上帝之眼"：数据透明化让决策层直接对接一线，信息损耗几乎为零，无需中层；

2. 中间层价值被消解：AI接管流程操作与分析工作，中层传统职责只剩分析与指导，失去存在必要性；
3. 钉钉实践：内部已无传统财务、设计师和人事团队，员工直接对接一线流程，成为"超级个体"。

我也听过另外一种观点是：未来中层会更重要，因为他们知道怎么样分配任务，派人干活，所以很适合去指挥。

这两种说法看似相反，实则指向一件事，那就是：只传递信息的人可以不要了，而能够有承担责任、协调产出和交付的人，到哪里都不会失去工作。

又要马儿跑，又要马儿不吃草

做产品贪心，目标 ABCDEFG 都想满足，对员工同事也贪心。不过这点说多了有点像抱怨，就不多说了。

只说一点，早前过完年回来刚推悟空的时候，还推我们所有任务都尽量在悟空里跑，上个月突然开始拉紧了算粒的口袋，申请算粒要逐层向上申请审批了。

申请打到上面，上面和我聊，是我不会写skill的事。

首先，限制用量是限制试错，限制试错就是限制创新。我觉得现阶段还不是限制员工用量的时候。

其次，AI 做出来是服务人的。我们努力适应 skill 写法，有财务上的必要性，没有产品设计上的充分性。毕竟我们服务的客户，只会比我们更不会用 skill。

再次，算粒消耗过高，是悟空 cache 收费做得有问题导致的比例更高，而根本不是 skill 的问题。悟空 context 管理的进步空间很大。其实我和其他同事之前翻过它的 trace，让它对比自己和 claude code：

分维度评级（对标 Claude Code）

能力维度	该 agent 团队	Claude Code 团队
把功能跑起来	B+	A
流式 / UI / 工具管线	B	A
把 KV cache 当成数据结构对待	**D**	**A+**
线上观测纪律	**D+**	**A+**
用类型 / API 设计防脚枪	C-	A
认真读 Provider 文档	C	A
把 LLM 调用成本当成一等指标	**D**	**A+**

合计：不错的特性工程师，弱在 LLM 系统层。看起来像一支从 web / 后端转过来的团队，把 LLM 当成另一个 HTTP API 来调，没有意识到这个 HTTP API 的服务端有一个昂贵的、有状态的、由客户端字节级控制的 KV cache。

在五月的某个版本，system prompt context管理突变，变得和claude code相似度很高，和老版完全不同，有点像完全替换。不过，底层换过来，cache 缓存却还按照 100% 收费，算粒消耗才那么难看。

不说底层工程的 infra，就连定时任务的设计，到现在好像还是在同一个窗口里面，context 不断累计。我三月份就提给悟空产品开发的问题，到现在六月份了也没优化（

体验报告

使用悟空的九个痛点（疼痛程度降序））。

至于写skill的能力——我赞同对产品经理要有要求，我对自己也会有高要求。不过人的时间是有限的，把时间分配在 skill 成本优化，还是客户服务，还是交互设计走查——要排优先级的话，我觉得针对 skill 单例的成本优化没那么高。

另外，其实我也理解财务方面的考虑。5 月 14 号的时候，微软也开始取消部分员工的 Claude Code 内部许可了。但人家一半是出于成本考虑，一半是为了推自家重金自研的模型和工具啊。

咱们自家不是有 Qwen 吗？怎么 Qwen 也要抽成限制算粒呢？

欸，看看人家 Qoder。

思之令人发笑。

流水线逻辑能否管理创造性劳动

四月份的时候，有一阵子，每天中午一点半午休刚结束的时候，无招会在工区巡视。

有回我也撞着了，只见阳光透过玻璃窗折射进来，在他头顶抛下一个天使的反光圈。他皱着眉，像一颗闪亮的金色飞贼，在工位间闪动游走，后面陪同巡幸的 HR 和 leaders 们骑着光轮 2001 都追不上。他游走到一些同事的身后，浑然不觉的同事就这样被他抓着「玩」微信了。

金色飞贼的嗡鸣声愈发尖锐，这位同事的 SM 被要求手写一份检讨书发到群里。

雅典学院 v.s. 西点军校

钉钉是一座巨大的军营，一座西点军校。

西点军校训练服从和纪律。它适合打仗，适合明确目标下的执行，降低命令的信息损耗。秦始皇花一十三年统一六国，统一的当年就通过最高法令一刀切式地完成了度量衡的统一。秦始皇在秦朝标准的基础上，直接就颁布了40字的强制诏书（「廿六年，皇帝尽并兼天下诸侯……法度量则不壹，歉疑者，皆明壹之」），直接把诏书刻在官定的秤砣（权）和量器上，分发到全国作为强制标准直接执行。

像度量衡这种规范，少量专家拍板就好，如果允许大家来回来讨论，二三十年不知道不知道论不论得完。

只是这种制度，真的适合做好 AI 产品吗?

与之完全相反的是雅典学院式的环境：容纳争论、游荡和奇想，鼓励师友之间辩论交流，大家可以「浴乎沂，风乎舞雩，咏而归」——它未必高效，却适合产生复杂判断；Google Deepmind和早期的 OpenAI 就属于这种。

前段时间看了一部开国时期的老电影，叫《姐姐妹妹站起来》，讲旧社会妓女在剥削压迫中觉醒自救。老鸨并不关心「女儿」的健康、情绪、尊严和命运，只关心有没有上钟接客，时间是不是高 ROI 地转化为对客产出。

从旧社会的余孽坟场中撇出来，一个普世的朴素判断是：生产者在怎样的环境中产出，就会在产品里植入什么样的基因，呈现给下游消费者。

残酷的养鸡场，养不出健康快乐好吃的白羽鸡。

同理，用流水线逻辑管理创造性劳动，产品也会带上流水线的气味：它能动能响，能给出进度，却缺乏灵气和温度。

服从性测试

在发心章我就提过，当时面试进入的流程并不轻松，要做族谱上钉的大作业，以及要过他本人的面试。某种意义上，让求职方先付出较高的沉没成本，是服从性测试的一环。但如果读者以为只有上山前需要纳投名状自证，就未免天真了。

挣扎试用期

钉钉的试用期是一道门槛，有好几个和我差不多同期来到钉钉进入 ONE 的同学，一个产品，一个运营，都没过试用期就直接走了。

试用期要交这样的三个个材料：

1. 阶段性工作证明；
2. 深度优化一个产品功能的客户共创服务报告；
3. 在钉钉的服务池里选择一个低分企业，或者自己拉一个新组织上钉，并把该组织服务到 V6 1000 分以上。

第一个比较基础。

第二个由第「用户」章，也可以窥知一二。

至于第三个，钉钉的 V6 1000 分企业是什么概念呢?

V6 是钉钉企业等级，门槛≥900 钉钉指数；1000 分是超满分头部水准，属于全平台顶尖数字化企业。全平台仅 5% 左右企业能冲到 900+，1000 满分企业不足 2%，大多是细分行业龙头、连锁品牌、规模制造 / 商贸公司。

不依靠特殊关系，用正常手段找到一个愿意配合的企业，并将它带到 1000 分，需要相当的运气，和起码 20 个小时以上的付出。我在找到愿意配合的企业这点上就付出了许多时间，打了很多无效电话，并且陌拜调研走访过不少周边小企业，可能额外付出在 30 个小时左右，都是用每周一天休息日的周六做的。

2. 线上公域流量测试（C端切入）

- **渠道覆盖：** 小红书、闲鱼、百度贴吧等高流量社区。
- **策略验证：** 测试了"SaaS选型咨询"、"行业痛点解决方案"、"反向调研"等多种内容触达方式。
- **结论：** 泛流量平台对B端决策者的转化链路较长，单纯的工具属性难以形成快速的组织裂变。



2. 线下实体业态走访（B端直面）

- **样本选取：** 选取HelloWorld Coffee、村上千层蛋糕、立体射箭馆等具有代表性的小微实体店。
- **痛点挖掘：** 针对排班、考勤、进销存等高频场景进行推介。
- **反馈洞察：** 极小微组织（<10人）对"流程化管理"的接受度与数字化迁移成本之间存在显著倒挂，即时通讯工具（如微信）仍是其首选竞品。



走了 30 个小时的弯路之后，最后还是找了朋友的公司帮忙，非常不好意思地请求并拜托了很多人，一起在钉钉上完成了许多功能使用打卡，才刷足了分数指标。

做 V6 任务的时候，我常常走在路上或正在吃饭的时候，心里突然升腾起一种巨大的无意义感。为什么我要做这个？这究竟在考验我什么方面的能力？这些花出去的时间，能帮助我成为一个更好的产品吗？或许也会的，但绝对绕了大量的远路，以一种我在常态下很难主动希望和寻求的方式。

自驱才能顿悟

庞加莱是很多认知科学和心理学研究「灵感」时必讲的经典案例。他在自传里这样定义自己的许多数学：「逻辑是证明的工具，直觉是发明的工具。」

他在搞定富克斯函数的第一阶段后，卡在了第二阶段。于是为了放松，他决定离开家，去参加学校组织的一次地质考察旅行。在旅行中，他刻意让自己不去想任何数学问题，每天看山看水看石头。

就在旅行结束、他准备坐公交车回家时，神奇的事情发生了。当他的脚刚踩上公交车的第一级台阶时，没有任何征兆，他的脑海里突然跳出了一个完美的答案：他之前研究的富克斯函数，本质上和非欧几何（双曲几何）的变换是完全同构的。

当时庞加莱甚至没有停下脚步，如常地继续登车落座，和身边的朋友谈天说地，仿佛什么都没发生。直到几天后回到家里，他才把这个结论记录下来，完成了世纪证明。

创意型工作无法绕开灵感；有的秩序环境是一座灵感和顿悟的温床，有的则反之。

如果庞加莱被逼迫参加一些数学兴趣班，或者在他刻意让自己不去想任何数学问题、按照自己的节奏培养灵感的时候，被要求每个周末必须去校长办公室汇报数学研究进展，并承诺每日精进，很难想象事态的发展。

写及此处，不得不再次叠甲：

我承诺，我理解产品是商业行为，不是学术研究；我理解向用户交付、向资方和股东交代，是产品的本职工作。

不过，有明确目标方向的赶路型产品开发，和没有明确目标方向的探索创新型产品开发，底层终究是不同的；交付用户、交代股东的方式，以及协议的宽容度，也大有不同；即便宽容度一致，项目一号位具体怎么带队落成执行，也有天差地别的余地。

说到天差地别——其他友商究竟是怎么做的呢？我在过去的一年不到的时间里，观察到的市场竞争又是怎样的状态和格局？

军争第七

「军争」在这里指代「竞品」。之所以起这么个半吊子古章节名，主要因为《孙子兵法》第七章正好也是「军争第七」，映照有趣，也就直取不改了。

《孙子》云，「军争为利，军争为危。」争到先手，当然是利；争错方向，也会把队伍带进危地。AI 协同办公这一仗，中原逐鹿，谁都知道要快，但真正的分野在于：谁知道自己该先争哪一段路。

前六章写的都是钉钉内部。发心、定位、设计、用户、敏捷、秩序，像把一支队伍出征前的马匹、粮草、军令、士气、营帐都看了一遍。到了这一章，要看它走出营门帐外，面前究竟是什么战场。

《易传 · 杂卦》有言，「比乐师忧。」

比卦，是亲比相辅；师卦，是兴师动众。人和组织在一起，最好当然是相亲相辅，各成其事，所以乐；一旦到了列阵相争、彼此窥探、互相赶杀，就进入存量竞争，无疑忧患。

这一章写「军争」，但不是为了把竞品关系写成热血战争。协同办公这个行业很有意思：大家一面互相竞争，一面又在被同一个时代推着往同一个方向走。钉钉、飞书、企业微信、Google、Slack 都在争企业 AI 的入口，也都在被模型、客户、组织成本、数据权限这些共同问题牵着走。所谓军争，争到最后，仍然要回到一个很朴素的问题：谁能让组织里的合作变得更顺。

由于这是针对钉钉内部工作经历的复盘，我无意展开非常多的竞品分析，更多还是写这一部分和钉钉更密切相关的内容。

探马

我在钉钉工作期间，接到过两三次自称飞书 HR 的电话。对方上来还没有自报家门，开门先问第一句就是我是不是在钉钉工作。有时候即便我表达了最近没有在找工作的意愿之后，对方仍然提出有没有「行业交流」的意愿。

就最近，五月中，我又接到一个电话，自称字节的 HR，问我「同学最近是不是在看机会呀」。我说自己不在看机会，对方还是追问：团队多少人、怎么分工合作、怎么做 Agent 测评。

我说，我们都签了保密协议。她顿了一下，又问我的汇报关系是不是某某某。

我问：「这还是个面试机会吗？」

对方说不是，就是想问一些问题。

然后就非常直接地开口，问产品关系，问业务数据。

我当时都愣住了，说 hold on hold on，我们这个是不付费的行业情报收集吗？又问对方是字节哪个 BU 的。对方吞吞吐吐说是飞书。

虽然我觉得钉钉有许多可进步空间，虽然我也理解面试套情报是竞争里的常见策略，但我实在看不惯这种直白的贪婪和不聪明，所以我对飞书观感也不怎么样，尽管产品层面上我欣赏友商的很多设计。

事以密成

钉钉对内对外保密措施都很严。文初提到，我入职开始就参与了保密的「O 项目」。到入职的下午快晚上，我才知道「O」是「ONE」的缩写。

825 发布会前，是 ONE 最秘密的开发期。大多数人知道有秘密项目存在，但不知道项目里具体在开发什么。我们向外部团队要开发资源时，只能非常含糊地提出需求，却又不能给对方看完整界面。

结果是，一个很小的接口需求，也要来回对三遍，拉很多人。大家都在配合，也都很辛苦，可信息一层层损耗，很多沟通就变成了猜谜。

秘密项目有它的道理，但它不是靠「秘密」本身成事。 洛克希德的 Skunk Works 常被拿来证明，大组织里需要一块隔离区。P-80、U-2、SR-71、F-117 这些项目，确实是在高度保密、相对独立的小团队里做出来的。可 Skunk Works 的关键不只是保密，而是「小而完整」：目标清楚，授权集中，团队能在内部快速闭环，对母体组织的依赖被压到最低。它像一把短刀，刀身短，刃口利，握在一个人手里，因而能快。 秘密项目最怕学到皮，没学到骨。只学到「关门」，没有学到「小团队闭环」；只学到「少数人知道」，没有学到「少数人就能把事做完」。这样一来，门是关上了，接口还在外面；信息是藏住了，资源还要四处去借。项目越秘密，越需要别人配合；越需要别人配合，越不能把话说透。

当时我们向外部团队要一个接口，却不能把完整场景摊开；对方不知道用户在哪里触发，不知道页面长什么样，也不知道这个字段为什么非要这样返回。他当然可以配合，但只能按局部理解来配合。一个小需求对三遍，拉很多人，为了在盲人摸象的情况下达成共识，搞了许多乌龙。

另外，秘密项目的代价不只是「沟通成本高」。

更深的代价是，它重新分配了解释权。掌握全貌的人，可以判断方向；只拿到局部的人，只能兑现需求。中心的人讨论为什么，外围的人追问什么时候。大家仍然在同一个项目里工作，却没有站在同一张地图前。

事以密成，前提是密处能成。

如果密处不能自足，秘密就会从保护机制变成信损机制。

在这种信息秩序里，越靠近中心的人，越知道为什么；越在外围的人，越只知道做什么。中心变强，理解变窄。项目看上去更快，组织其实更依赖少数人的解释权。

这仍然是一个选择，不是一个问题。只是这就依赖掌握核心解释权的人，有非常强的能力和产品理解——秦始皇可以，秦二世不行。

保密是有必要性的。竞品会打探，供应商会流动，客户会比较，媒体会放大；一张截图、一段录屏、一个命名、一次提前泄露，都可能造成商业损失。

所以，真正要讨论的不是要不要保密，而是保密的边界画在哪里。

张小珺有一期播客，和广密聊全球大模型格局，对他们谈到 Anthropic 的一段，让我印象很深。大意是：Anthropic 在外部竞争上很谨慎，很多东西不轻易往外说；但在组织内部，又保留了很强的讨论和挑战空间。公开报道里也能看到相近的侧写：Anthropic 鼓励员工在 Slack 上直接挑战 Dario Amodei，内部有开放的长文讨论和 notebook 式分享；同时，它对外又用模型卡、安全政策和透明度页面，选择性地披露自己愿意披露的部分。

这或许可以给我们一个参照：好的保密，边界应该主要画在组织和外部世界之间，而不是一层层画在同一支队伍内部。

外部可以保密，内部要保共识。

外部不能让对手看见粮道，内部不能让同袍只拿到一段断简残编。

钉钉的保密不只挡住了竞品，也挡住了协同者。这种信息秩序会变成战斗力的一部分：一支队伍当然不能把军情贴到城门口，可如果营内的人也不知道自己要去哪里，密就不再只是密，也是额外的人力时间损耗。

战场改名：从协同办公到企业 AI 入口

过去一年，协同办公的竞争已经换了名字。

文档、表格、会议、搜索、机器人还在打，但它们只是局部战斗。整场战争变成了：谁能成为企业 AI 的第一入口。

这个「入口」不只是 App 首页。它有三层意思：

1. 用户每天从哪里开始工作；
2. AI 到哪里拿企业上下文；
3. 任务最后由谁调度、记录、执行、兜底。

国内御三家都在往这里靠。

钉钉在 2025 年 8 月 25 日发布 8.0，也就是 AI 钉钉 1.0。官方和媒体报道显示，这次发布包括钉钉 ONE、AI 搜问、AI 表格、AI 听记、DingTalk A1 等超过 10 款 AI 产品；无招在发布会上说，钉钉要以归零心态为 AI 时代打造一个全新的钉钉。

飞书在 2025 年 7 月「未来无限大会」发布和升级多维表格、知识问答、AI 会议、Aily、妙搭等能力。21 世纪经济报道引述飞书多维表格负责人信息称，多维表格月活跃用户接近千万；飞书还发布企业级 Agent 开发平台 Aily，以及通过 AI coding 帮企业搭系统的飞书妙搭。

企业微信在 2025 年 8 月发布 5.0，新版本推出智能搜索、智能总结、智能机器人三大核心 AI 功能，并把 AI 接入更多办公模块。企业微信负责人黄铁鸣披露，企业微信已接入超过 1400 万真实企业与组织，每天通过企业微信服务的微信用户超过 7.5 亿。

海外正常也在同一条线上争分夺秒。

Google 在 2025 年 10 月发布 Gemini Enterprise。Google 官方把它称作 workplace AI 的「new front door」，说它通过 Gemini 模型连接公司信息，帮助员工协作，并用 Agent 自动化任务；Google Workspace 也把 Gemini 编进 Gmail、Docs、Sheets、Meet、Chat、Vids 等日常工具里。

Slack/Salesforce 在 Dreamforce 2025 前后把 Slack 往 agentic OS 上推。Slack 官方说，Slack 是员工花最多时间的地方，因此适合把对话、应用和工作流统一起来；Salesforce 也把新 Slackbot 描述为 Agentic Enterprise 的 front door，可以在 Slack 内跨会话、日历、CRM 等查找、提问和行动。

用表格概括一下这个 「AI 协同办公」的战场：

阵营	旧地形	AI 新打法	最想拿下的东西
钉钉	组织、流程、审批、IM、日程、会议	ONE、AI 搜问、AI 表格、AI 听记、A1、悟空、Agent OS	企业工作入口和执行入口
飞书	文档、知识库、多维表格、会议、项目	知识问答、多维表格 AI、Aily、妙搭、AI 会议	知识工作和业务系统入口
企业微信	微信客户连接、社群、私域、上下游	智能搜索、总结、机器人、智能表格	客户经营和社会化协同入口
Google	Gmail、Docs、Meet、Drive、Cloud	Workspace Gemini + Gemini Enterprise	全球办公套件里的 AI 总入口
Slack/Salesforce	Slack 对话 + Salesforce CRM	Slackbot、Agentforce、Enterprise Search	对话现场里的 CRM/企业 Agent 入口

不难发现，AI 没有凭空造出新地形，而是由选手们各自依托各自的优势，在旧地形重新开战。

钉钉的底盘，是组织确定性。

组织架构、上下级、审批、考勤、DING、已读未读、任务推进，这些放在个人体验里未必讨喜，但在企业管理里很要紧。钉钉早年的胜利，靠的就是把「我说的话对方看见没有」「我交代的事有没有往前走」做成系统。

所以钉钉做 AI，很自然会走向主动服务和组织推进。ONE 是工作秘书，AI 搜问是全局搜索，AI 表格是业务自动化，AI 听记和 A1 把会议与声音收进系统，悟空则把「你说，我干」推到执行层。钉钉官网对悟空的定义也很直白：企业级 AI 工作平台，让 AI 从「会回答」进化到「能交付」，用户说需求，它调工具、用技能，把工作推进到结果。

飞书的底盘，是知识和结构化协作。

飞书最强的地方，不是催人，也不是触达客户，而是把复杂工作整理得漂亮。文档、知识库、多维表格、会议纪要、项目协同，围绕的都是一件事：把人的工作沉淀成结构。飞书知识问答官方说明里写得很清楚：它能根

据用户在飞书上有权限访问的消息、文档、知识库等资料给出回答和进行内容创作；多维表格则被飞书定义为兼具表格轻盈和业务系统能力的新一代效率应用。

企业微信的底盘，是微信连接。

企业微信不必像钉钉和飞书一样重新证明自己是入口。它天然站在中国企业最重要的外部关系旁边：客户微信、社群、私域、门店、服务、上下游。企业微信 5.0 的智能搜索、智能总结、智能机器人，都是把 AI 放进原来的客户沟通和办公协作里。

Google 的底盘，是 Workspace。

Gmail、Docs、Sheets、Meet、Drive、Chat、Vids、NotebookLM 这些工具每天就在办公流里。Google 的做法很自然：一边把 Gemini 编进这些日常工具，一边推出 Gemini Enterprise 作为统一 AI 平台，让企业在同一个安全环境里发现、创建、共享和运行 Agent。

Slack 的底盘，是对话现场和 Salesforce 数据。

Slack 的价值不是「聊天」两个字，而是企业里大量工作的现场感：某个 channel 里发生过什么，谁参与了，文件在哪里，客户数据在哪里，销售和服务流程推进到哪一步。Salesforce 把 Slackbot 说成 Agentic Enterprise 的 front door，正是因为它可以在 Slack 内跨 conversation、calendar、CRM 等完成搜索、提问和行动。

这五路打法，区别很清楚：

- 钉钉想替组织把事往前推；
- 飞书想替知识工作者把事理清；
- 企业微信想让企业和客户之间少一点摩擦；
- Google 想把 AI 放进全球办公套件的每一条缝；
- Slack 想让 Agent 住进企业对话和 CRM 数据之间。

旧基因决定新打法。

发布会里的机锋

钉钉很重视发布会。

无招回归后，每次发布会都像一次改元。一个 logo 像一个年号：从魔法棒，到 ONE 的太极标，再到悟空猴头。一朝天子一朝臣，每换一次符号，就重讲一遍故事。

这种打法有气势。大组织需要旗帜，需要阶段感，需要一个全体成员都能对齐的方向。发布会像军令，可以把组织的注意力一下拉到同一面旗下面，也可以对外宣告：钉钉已经进入 AI 时代。

问题是，钉钉发布会往往太大、太满、预售感太强。

钉钉 8.0 发布会时，无招用 200 多页 PPT、历时 2 小时，一口气展示了十余款 AI 产品，包括 ONE、AI 听记、AI 搜问、AI 表格和 DingTalk A1；2025 年 12 月 23 日钉钉 1.1「木兰」发布会上，钉钉又发布 20 余款 AI 新品和 DingTalk Real，并把叙事抬到「全球首个为 AI 打造的工作智能操作系统：Agent OS」。

产品发布得多，当然显得有声势。可用户在发布会上看到，回到钉钉里可能没有；几个月后上线，形态又常常和台上不完全一样。前线销售最痛苦，因为他们要把发布会上的愿景翻译成客户能买的东西：怎么开通、多少钱、卖给谁、效果怎么保证、出了问题谁兜底。

ONE 就很难卖。它是入口，是叙事，是未来工作方式的一部分。可销售站到客户面前，客户要的是 SKU、权益、场景、交付边界。发布会能振奋人心，销售现场只认能不能用、能不能卖、能不能交付。

飞书在这件事上打得很准。

2026 年 3 月 19 日飞书春季发布会，外部报道标题就叫「全程 Demo」。报道说，这场发布会全程纯 Demo，没有 PPT，一共发布 8 款产品，包括飞书 aily、aily 专业版、飞书妙搭、多维表格 AI 搭建、AI 同传、会中 AI 纪要等；谢欣开场解释这个形式时说，AI 产品每天都在迭代，传统方式跟不上了，「Slides are cheap, show me the demo」。

这句话很损。它表面讲飞书自己的发布会形式，放在竞品语境里，几乎就是对钉钉的嘲讽：PPT 很便宜，演示给我看。

这里不是 PPT 和 Demo 的审美差别，背后是两种打法。

钉钉用发布会调动未来。

飞书用 Demo 兑现当下。

钉钉像改元，飞书像验货。钉钉在台上给组织换年号，飞书把电脑摆出来让用户看能不能跑。

但这一仗也不能只写飞书高明、钉钉狼狈。无招有反击，而且反击很有意思。

飞书曾经用「钉钉只有打卡数据，做不好 AI」这一类话术嘲讽钉钉。这个说法不一定有正式公开出处，但它确实符合飞书一贯的竞争姿态：飞书强调文档、知识库、多维表格和结构化沉淀，天然会质疑钉钉的上下文质量。钉钉有很多组织行为数据，但知识资产、文档结构和语义沉淀，确实不如飞书顺手。

无招反过来提出：到了 AI 时代，不该还让人写文档。

人本来就是说话的动物，真实工作大量发生在会议、口头沟通、现场讨论里。与其逼人先整理成文档，不如让大家开会、口述、讨论，把声音全部进入听记，再让 AI 整理纪要、文案、任务和后续动作。

于是钉钉推出 AI 听记和 DingTalk A1。钉钉 8.0 官方材料把 DingTalk A1 说成支持智能听写、安全加密的语音智能助手，也把 AI 听记定位为适用于会议记录、任务跟进等工作场景的语音智能工具。

这段交锋很精彩。

飞书相信，文档、表格、知识库是 AI 的土壤。

钉钉反手说，真实工作本来就不从文档开始。

飞书先沉淀，再智能。

钉钉先捕捉，再整理。

飞书像图书馆，钉钉像录音棚。图书馆清楚、安静、可索引；录音棚嘈杂、鲜活、贴近现场。谁更适合 AI？很难一句话说死。飞书的优势是知识资产清楚，钉钉的机会是更接近日常工作发生的第一现场。

这一仗，至少打得有来有回。

Agent：远方和泥地

Agent 是所有平台的远方。

钉钉有悟空和 Agent OS。

飞书有 Aily、妙搭和智能伙伴。

企业微信有智能机器人。

Google 有 Gemini Enterprise。

Slack/Salesforce 有 Slackbot 和 Agentforce。

钉钉悟空官方页面写得非常直接：「你说，悟空干」；它被定义为企业级 AI 工作平台，让 AI 从会回答变成能交付，能够调工具、用技能，把工作推进到结果。 2026 年 3 月，媒体报道钉钉发布悟空时也提到，悟空作为独立应用开启邀测，也将内置到钉钉 AI 2.0 中；钉钉重写底层代码，进行全面 CLI 化改造，让悟空能够原生操作钉钉上千项能力，目标是实现「沟通即执行」。

飞书 Aily 走的是企业智能体平台路线。飞书官方说明里，Aily 可以通过 MCP 协议连接企业知识与业务系统，内置工具与服务市场，让智能体懂业务；2026 年飞书 aily 自定义智能体升级，也强调帮助企业为报销专员、项目助手、HR 答疑机器人等业务场景打造专属 AI 数字员工。

Google 的 Gemini Enterprise 更像企业 AI 总平台。Google Cloud 官方说，Gemini Enterprise 通过一个直观聊天界面，作为 workplace AI 的 single front door，并统一模型、Agent 创建、预置 Agent、数据集成、安全设置和生态伙伴等核心组件。

Slack/Salesforce 这边，Slackbot 是最清楚的参照。Salesforce 官方客户故事直接说，Slackbot 是 Salesforce Agentic Enterprise 的 front door，员工可以在 Slack 里跨 conversations、calendars、CRM 等搜索、提问和行动。TechRadar 也报道，Slackbot 集成 Salesforce、Slack 数据和 Google Drive、calendar apps 等第三方服务，能检索文档、总结项目进展、安排会议、识别遗漏优先级。

Agent 真正难的地方不在聊天，而在做事。

它要知道上下文，要有权限，要能调用工具，要知道失败以后怎么收场。只会生成一段话，不叫 Agent；能在组织系统里安全地推进一件事，才算摸到门槛。

Salesforce 研究者 2025 年发布的 SCUBA benchmark，也说明企业任务自动化并不轻松。这个基准用 Salesforce 沙盒环境评估客户关系管理工作流中的 computer-use agent，覆盖平台管理员、销售和服务人员等角色；论文摘要提到，在 zero-shot 设置下，开源模型驱动的 computer-use agents 成功率低于 5%，闭源模型方法最高也只有 39%，加示范后可提升到 50%。这说明企业软件里的 Agent，远比发布会 demo 难。

这恰恰是钉钉的机会，也是钉钉的风险。

机会在于，钉钉有组织、权限、审批、日程、会议、IM、低代码、表格、阿里云和通义。它比许多纯 Agent 创业公司更接近真实企业系统。

风险在于，钉钉容易先把旗子插到 Agent OS 上。小闭环还没跑稳，国号已经立了。

广积粮，缓称王。

消息能不能处理，日程能不能闭环，会议能不能变待办，表格能不能接业务系统，知识问答能不能可靠，这些小仗没有赢，Agent OS 就容易飘在空中。

望舒(书)行动

2026 年 4 月 2 日晚上，SM 突然通知所有 SM 和 PD 12 点前不许下班，看飞书那栋楼几点熄灯。因为一份传说是飞书供应商写的飞书 Aily 对比钉钉悟空的分析报告，送到大老板面前，条条证据直指钉钉 AI 体验在各项具体工作任务中的欠缺。

我留到十点半多，意识到自己只是为了呆着而呆着。和三月份的每一个周末一样。是的，整个三月份要求全员加班，周六日两天都要来。

看街对面的竞对几点熄灯，仿佛看见对方的灯还亮着，就能反推出对方的意志还在燃烧；再推演到只要我们的灯也不灭，就能抵消战略上的不确定。

什么样的人的想象，惟在这一层能够如此跃进?

可是竞对在电影院站起来，就代表我们也该起身内卷对抗吗？何况，那份报告含沙射影踩了一脚悟空城门的火，竟然燎到基础线产品的池鱼。以及，通过表现努力来安抚焦虑的后果，是员工更不良的工作状态，和更低迷的士气，还给了竞对很好的上下离间的抓手。

我感到一种强烈的无意义感，向 SM 请假回家了。

4 月 25 日前后，不少社媒传出通稿，谢欣内部呼吁飞书员工早点下班。虽然飞书也是加班重镇，评论区一片调侃，但这条新闻在钉钉内部也算半阙楚歌了。

晚点 LatePost 在 V4 发布前夕采访了 Deepseek，后者不加班、不打卡、没有明确的绩效考核，平日多数成员会在 6 点~7 点左右离开公司。梁文锋表示：一个人每天能高质量工作的时间很难超过 6~8 小时。加班疲劳下的昏庸判断反而会浪费宝贵的算力资源，得不偿失。

这个判断听起来像常识，但在很多组织里，这种常识是非常稀缺的。

https://mp.weixin.qq.com/s/bYZrKp48Y7EpsU8_vd6TcQ

西南航空：战略在于放弃

战略不是把所有好东西都做一遍。真正的战略，本质上是放弃的艺术。

在商业史上，西南航空是一个绕不开的名字。这家成立于 1971 年的廉航公司，在航空业这个以烧钱著称的行业里，创下了连续 45 年盈利的惊人纪录——即便在 911 事件后全行业哀鸿遍野的 2001 年，以及 2009 年金融危机的寒冬中，西南航空依然保持盈利。它从不试图讨好像有所有乘客，从不追求航线网络的无所不包，从不提供高端舱位和奢华餐食。然而，恰恰是这些「不做什么」的选择，成就了它独一无二的竞争壁垒。

西南航空放弃了什么?

首先，放弃机型复杂度。 西南航空的机队只有一种机型：波音737。几十年来，这家公司从未像传统航司那样运营波音、空客、庞巴迪等多个品牌、多个型号的飞机。单一机型的战略选择，使得维修保养、技术手册、工具备件都得以高度标准化，飞行员培训成本大幅降低，调度编排极大简化。与此同时，波音 737 出色的燃油效率帮助西南航空压低了运营成本。

第二，放弃枢纽中转网络。 与传统航空公司采用「枢纽-辐射式」（hub-and-spoke）模式不同——乘客需要从一个小城市飞往芝加哥、亚特兰大这类大型枢纽，再转机去往最终目的地——西南航空选择的是点对点直飞模式。飞机从 A 城直达 B 城，没有复杂的经停和中转。这种做法让乘客免去了中转的麻烦和延误风险，也让航空公司减少了因航班延误而产生的连锁反应——一架飞机延误不会波及整个网络。

第三，放弃复杂餐食和客舱分级。 西南航空从来不在飞机上提供正餐或热食。它的客舱不分头等舱、商务舱、经济舱，所有人坐在同样类型的座椅上，没有座位预留，不提供升级机会。公司有个著名的判断：「没有人是为了吃的而坐飞机的。」与其在餐饮和舱位分级上投入大量成本，不如把价格降到最低。

第四，放弃讨好所有乘客的野心。 西南航空从不试图满足每一个乘客的所有需求。它不设头等舱，开放选位机制让一些习惯传统服务的乘客感到不适；它不做航班中转，对一些需要复杂行程衔接的商务旅客不够友好；它的机上服务极其简约，长途旅客可能觉得「简陋」。但西南航空恰恰是通过放弃这些「看似好」的服务，才赢得了一群对价格敏感、追求效率、愿意牺牲「过度服务」的忠诚客户。 那么，这样大刀阔斧的放弃又带来了什么回报?

西南航空这种主动放弃的姿态，绝非自废武功式的自我削弱，而是有策略地「断舍离」，换来的是三大战略优势。

一是极低的成本。 单一机型加高超的飞机调度能力，使西南航空的飞机日利用效率维持在 11 小时左右，而同行大约只有 9 到 10 小时。其航班过站周转时间平均为15分钟，而其他航空公司可能需要两到三倍的时间来完成同样的工作。这种极致的效率将运营成本压缩到每座位10.2美分，而同期联合航空的成本高达 15.7 美分。

二是稳定一致的客户体验。 乘客搭乘西南航空时，可以准确预知将要经历什么：一定是波音 737，一定没有正餐，一定不安排座位，一定可以免费托运两件行李。这种一致的体验降低了乘客的心理预期管理成本，也让西南航空在 J.D.Power 的经济舱客户满意度排名中连续多年位居榜首。调查显示，西南航空的净推荐值（NPS）高达 48 分，接近 60% 的旅客愿意向他人推荐这家航司。

三是持续的盈利能力。 西南航空在这些放弃的支撑下，从成立第二年起持续盈利了 45 年，这在全球航空史上是绝无仅有的奇迹。2001年航空业在911重创下大幅亏损、2009年金融危机全行业哀鸿遍野时，它依然屹立不倒。

话说到这里，还是要回应本章的核心：竞品竞争。如果说西南航空的「放弃」是一条清晰的战略选择，那为什么其他航空公司不能复制?

表面上看，其实几乎每一家传统航司都曾尝试推出自己的廉价子品牌——美国航空有「美航快线」，联合航空曾推出「Ted」，达美航空有「Song」——试图以廉航模式对抗西南的冲击。但这些模仿最终大多无疾而终。根本原因在于：这些航司在尝试「加法」的同时，却不舍得做真正的「减法」 。

传统全服务航空公司的基因里，根深蒂固地保留着复杂的业务体系。它们运营着多种机型、多级舱位、全球中转网络；它们习惯于为高端客户提供奢华的休息室和精致的餐食，无法轻易割舍这些「高质量服务」带来的品牌溢价。即便推出了廉航子品牌，那些复杂冗余的业务线依然以独立实体的方式存在，母公司依然保留着复杂的 SKU 和高昂的成本结构，真正赚钱的核心业务依然被各种「增值服务」层层包裹。

西南航空最大的竞争壁垒恰恰在于它「不做的事情」——传统航司哪怕也想做低成本，但在未拥有国际航线、商务舱与宽体机队等规模与产品优势的前提下，简单抄袭战术会导致品牌承诺与产品变化之间的深层撕裂。换言之，传统航司无法像西南那样「干净利落地放弃」，正是因为它们的商业模式和历史包袱不允许它们彻底简化。

即便那些专注廉航模式的模仿者——柏林航空、WOW 航空、尼基航空等——也都纷纷折戟沉沙。它们虽然「形似」西南航空，但在流程、文化、人才、管理执行力等维度上远未达到「神似」的境界，最终只得其表，不得其意。

总之，企业不可能同时在所有维度上做到最好，试图面面俱到的策略往往导致面面平庸。真正的战略优势，不是来自于你做了什么，而是来自于你敢于不做什么。西南航空放弃了复杂的SKU、豪华的舱位和丰富的餐食，换来了极致的成本优势、清晰一致的品牌体验和跨周期的盈利韧性。

有时候想想我们总是满打满算的发布会，钉钉最缺的，恰恰是放弃。

总结：看得远，打得急

最后，评估钉钉，要公平一点。

钉钉没有看错终局。企业 AI 的核心，确实是组织上下文、工作流、权限、入口和执行闭环。钉钉手里也确实有很多资产：IM、组织架构、审批、日程、会议、表格、低代码、阿里云、通义模型和大量企业客户。

只是问题在于太早用终局叙事组织中局战斗。

无招回归后，钉钉每一轮都有新旗帜。从魔法棒到 ONE 的太极标，再到悟空猴头。一个 logo 像一个年号，每次改元，都要组织重新站队，重新解释，重新相信。

这有好处。大组织需要旗帜，需要阶段感，也需要让内部知道「现在打哪一仗」。

只是改元太频繁，也会伤人心。

中国历史上，最爱改元的皇帝之一是武则天。她在位时间并不算长，却频繁更换年号。每一次改元，往往都对应一次祥瑞、一场政治转折、一次合法性宣示，或一次对佛教、道教、天命符号的重新调用。年号在她那里，不只是纪年，也是政治自我命名：我要告诉天下，如今又换了一个天命，又进入一个新局。

这种做法短期内有用。它能制造气象，重启叙事，让人感觉王朝一直在更新、在受命、在转身。可改元太频繁，也会让时间本身失去稳定感。对民间来说，年号本来是记账、立契、行文、报税、往来的公共坐标；坐标频繁变动，普通人并不会因此更理解天命，只会觉得朝廷的意志又压过了日常生活。

发布会太大，前线接不住。预售感太强，用户信任会被透支。战线太长，团队和客户都难形成清楚记忆点。用户迁移台阶不够，很多东西从愿景直接跳到入口。

旧时代的强管理手感，和 AI 时代需要的个性化、成本感、边界感、失败兜底之间，还没有完全接上。

钉钉看得远，也打得急。

它最有希望的地方，是它真的在企业工作流里；它最危险的地方，是太想一次性宣布自己就是未来。以友商为镜吧，飞书的锋利，在于它抓准了 AI 时代开发者需要什么；企业微信的聪明，在于它不打扰旧路；Google 和 Slack 的成熟，在于它们把 AI 放回用户原来的工作流，再慢慢抬高平台。

「军争之难者，以迂为直，以患为利。」看见远方以后，仍然有耐心先修脚下的路，这才是真章。

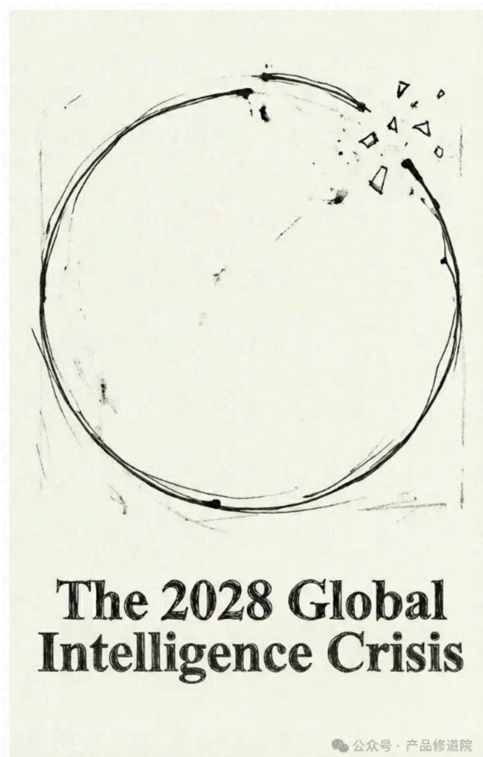
长期第八

26 年 2 月中旬，一篇叫《2028年全球智能危机》（THE 2028 GLOBAL INTELLIGENCE CRISIS）的长文刷屏了。紧邻的周一，道指暴跌820点，纳斯达克跌1.1%，软件板块集体跳水——很多人说这篇文章是催化剂之一。

文章模拟两年后的口吻，平静地叙述了一个AI太过成功反而导致经济崩溃的恐怖未来：白领大批失业，消费螺旋下行，房贷市场崩盘，标普500从8000点暴跌38%至3500点。

写得确实好，好到好多人读完感慨「脊背发凉"。

直接刷爆外网，文章在X上获得了超过1600万次浏览。《大空头》原型、传奇投资人Michael Burry亲自转发，配文："And you think I'm bearish."（你们还觉得我悲观吗？）



Citrini Research

THE 2028 GLOBAL INTELLIGENCE CRISIS

A Thought Exercise in Financial History, from the Future


CITRINI AND ALAP SHAH
FEB 23, 2026

5,640 83 1,217 Share

Preface

What if our AI bullishness continues to be right...and what if that's actually bearish?

What follows is a scenario, not a prediction. This isn't bear porn or AI doomer fan-fiction. The sole intent of this piece is modeling a scenario that's been relatively underexplored. Our friend Alap Shah posed the question, and together we brainstormed the answer. We wrote this part, and he's written two others you can find here.

Hopefully, reading this leaves you more prepared for potential left tail risks as AI makes the economy increasingly weird.

公众号 · 产品修道院

不过相比起一份站得住脚的逻辑推导，它更像一篇精心构建的恐怖小说。最大的问题在于文章的静态史观：作者似乎只看到了AI在摧毁什么，而完全没有考虑人类会创造出什么新东西来填补空缺。而过去200年的历史告诉我们，人类每次都会。

到了我们的最后一章「长期第八」，应该把镜头从 ONE、钉钉、无招、用户、发布会和竞品身上慢慢挪开，挪回到做事的人身上。

前面几章写了很多产品的事：设计怎么成立，用户怎么买单，秩序怎么塑造。写到这里，如果仍然只问「怎样把产品做得更好」，就又回到了旧路上。人还是工具，经验还是燃料，身体还是成本，痛感还是被提炼成方法论的矿石。

本章首先想问的不是公司、产品怎么长期，而是我们作为从业者怎么长期。

一个人来到一家公司，进入一个项目，跟随一位老板，赶一场发布会，参与一场军争，当然可以全力以赴。年轻时总要见识火的温度，也要在某些地方学徒、下注、冒险。问题在于，人不能永远以「先把这一阵熬过去」的方式生活。项目会换，年号会改，logo 会翻新。留下来的，是身体、判断、写作能力、亲密关系、对世界的兴趣，以及一个人还愿不愿意相信自己的感受。

长期是人心里的一本账：什么值得押注，什么只会辜负；什么促进本领，什么徒增痛苦。

哀我人斯，于何从禄？瞻乌爰止，于谁之屋？

这章不谈「诗和远方」，只谈「附近」，只谈对人的尊重和爱护。

人是目的，还是手段

什么工作养人，什么工作耗人

判断一份工作对人是滋养还是消耗，要看它最后把人带到哪里去：是让人练成技艺、养好身体、健全尊严，还是只让人变得疲惫浑浊、学会应付了事。

从反馈机制、体内激素情况等维度分析，养人/耗人型工作，大抵可以这样区分：

维度	养人型工作	耗人型工作
核心本质	身心合一： 身体在付出，心灵在吸收能量。	身心分离： 身体在机械运作，心灵在强烈抗拒。
自主权	高控制感： 节奏、方向、结果大致由自己掌控。	低自主性： 身不由己，像螺丝钉一样被系统推着走。
反馈机制	即时且确定： 付出能立刻看到成果（如干净的房间、做好的饭菜、长出的新芽）。	延时且虚无： 成果不可见，或毫无意义（如走形式的PPT、没有结论的扯皮会）。
体内激素	分泌 多巴胺（成就感）与 内啡肽（内心的平静与愉悦）。	持续分泌 皮质醇（压力荷尔蒙）与 肾上腺素（焦虑与防御）。
典型场景：体力	园艺、烹饪、做手工、洒扫庭除、适度运动。	流水线式的机械重复、超出身体负荷的被迫苦役。
典型场景：脑力	自发驱动的创作（写作、画画、解决真正感兴趣的技术难题）。	防御性内耗（办公室政治、揣摩人心、为不合理的指标编造数据）。
典型场景：情感	发自内心的利他行为、滋养彼此的深度倾听与沟通。	有毒关系中的「走钢丝」、被迫戴上面具的迎合与假笑。
边界感	边界清晰： 阶段性结束，能让人彻底放松和抽离。	无边界蔓延： 随时待命，神经系统像绷紧的橡皮筋无法松弛。
劳动后的状态	身体虽累，精神充盈，睡一觉就能满血复活。	慢性枯竭（Burnout）： 越睡越累，精气神被抽干，对万事冷漠。

我在 ONE 封闭开发期间，工作日的工作时间通常是从早上 9 点一直到晚上一般 11 点半过之后。有时候夜里等发版或者开会到一两点，虽然不是天天如此，但也可以说习以为常。周日固定需要工作，时间上稍微好一点，早十晚八。周六一般不来，但偶尔突击月的时候也得来。我最夸张的一个月，整个月只休了一天半，除去这一天半，打卡可以体现的日均工作时长是 15 个小时左右。

后来 ONE 不再在钉钉的焦点位，作息才稍稍好转。再后来到今年 4 月，我被调动到负责钉钉基础线日历日程的相关工作，才更好一些；不过仍然，工作日早九晚十，周日下午也有固定产品汇报。

去年至今在钉钉工作过的同学可能都知道其中滋味。

嗟乎！一人之心，千万人之心。「钉」爱奋苦，人亦念其家。奈何取之尽锱铢，用之如泥沙。

只是这样的环境，真的能做出理想的 AI 产品吗?

就像在「秩序」章所论证，用流水线逻辑管理创造性劳动，恐怕是失能的。

我们 C6 四楼中庭的项目室里，四十多个人坐在一个大房间，产品、开发、设计挤在一起，晨会、晚会都在这里开。四五排桌椅排得很密，空气不怎么流通，一有人生病，周围很快倒一片。那时大家都忙，忙到空气也像公用资源，被一遍遍吸进呼出，带着咖啡、外卖、疲惫和感冒药的气味。

一个组织如何回答「人是目的还是手段」，不在口号里，在这些小动作里。项目室怎么安排，休息怎么评价，请假怎么打分，竞对压力怎么传导。如果一个组织嘴上讲 AI、创新、长期，现场却把人的身体、尊严、注意力当作随取随用的材料，它其实已经回答了这个问题。

人当然可以把自己交给一件事。我不反对辛苦，也不反对高压。好的工作常常不轻松，真正学本事也很少舒服。做产品像做菜、做手工、做手术，都有一段学徒期。后厨里火热，刀快，师傅骂人，节奏紧，都可以理解。可学徒不是柴火。柴火烧完就算，学徒要长手艺。一个人如果在一段工作里只被燃烧，没有长出判断、技艺、眼界和更稳定的内心，这段燃烧就要重新估价。

身体不是项目资源，敏感不是可随意磨损的零件，判断力也不是无限透支的账户。一个从业者为长久计，第一件事就是承认自己是带着生命进场的，不是带着无限工时进场的。

快就是慢，慢就是快

ONE 收官不做以后，我开始负责钉钉日历日程的基础产品。这段时间所有基础产品都在开展走查竞对差距并补齐的工作。

「现在都有 AI 了，你们一周做完，一个月全面超越竞对。」

可是，不说竞品不会停工一个月静态地等待我们，就说以流水线的速度，要求艺术品的生产，这件事真的成立吗?

前文提到，AI 做服务或 C 端交互型产品，本质上是对奢侈商品或服务的向下放权。

爱马仕生产一条丝巾需要两年。不是流水线躺两年，而是一个图案从被选中到分色、制版、配色、试印、印制、卷边、上市，整个周期拉得很长。复杂图案要拆成几十个颜色层，每一层对应一次工艺判断。它慢，不是因为低效，而是因为价值本就包含审美判断、材料理解、工艺传承和反复校准。时间被收进了作品里。

这对做 AI 产品也有启发。公司当然要求快：窗口期短，模型迭代快，竞品动作密。问题在于，快分层次。能快的地方要快——原型、验证、反馈、低风险试错。可有些地方一味求快，最后更慢。比如用户心智没想清楚就发入口，权限成本没算清就包装 Agent，业务场景没打穿就堆发布会。表面抢了时间，后面要用返工、解释、客服、信任来还债。

这不是替低效辩护。真正有价值的慢，要能说清它在换什么：换更准的用户判断，换更稳的系统地基，换更少的返工，换长期可复用的能力，换用户愿意再来的信任。说不清换什么的慢是拖延，说得清才是长期投资。

所以长期要争的不是快或慢，而是哪段时间该省、哪段该投。低价值重复交给 AI，高价值判断留给人；可规模化的个性化交给系统，需要审美、责任和取舍的地方留出时间。这样的快不乱阵脚，这样的慢不耗心态。

不得不单独拎出来提一下健康问题

在 ONE 项目期间，我晕倒过两次，第二次同事打 120 叫救护车把我送去浙一急救了。

核心原因可能都是过长的熬夜工时和项目室的密闭环境。

第一次是深夜汇报过无招后，回项目室继续给开发评审需求。其实那个时候也才 12 点多快 1 点，不过已经连续熬夜了很久，又大量服用咖啡，导致精神状态不怎么样。当时是某个快速起身，突然就像电脑蓝屏一样，眼前一黑。后来同事告诉我，我砰一下倒下去了。不过好在只是短暂蓝屏了两三分钟，我自己又重启醒过来了。

那次后来都没去医院。因为我自己大概感觉可能是低血糖，低血压之类的毛病，去内科复查，大夫最多让我多休息，但那是我做不到的。

第二次是在一个理论上说比较健康的时间，晚上 7 点多。我只是如常在位置上写文档。几个开发吃饭回来，和我打了个招呼。我正要笑回，突然感觉心脏一紧，呼吸变得困难起来。

开发可能看我表情变了，有点疑惑地看着我，又面面相觑。当时我感觉，从心脏到左半边身体的感受，突然被加强了许多倍，四肢和脸上的皮肤开始发麻，声带也有点僵直不受控制，但还是挣扎地解释：「我感觉有点不舒服...」不过话没说完，就感觉心脏重重地一跳，一股更加汹涌的麻木感就从左胸往周围蔓延开来。

周围几个同事都有点紧张地过来问我怎么了。我想转身告诉他们我现在的情况，但声带好像已经变成了一块死肉，已经什么都说不出口了，于是我只能徒劳地摔在了地上。

一直感觉项目室的空气差，没想到靠近地面的空气会更糟糕。我侧倒在地上，脸贴着灰色静音地毯，浑浊的空气混着地毯有些发霉的潮气钻进鼻腔。

「快打 120！」「把她扶起来」「去叫 HR！」「能不能扶啊？不能随便扶吧？」……

几个同事还是把我的上身扶起来，让我靠在其中一个女生的腿上，这时候我已经感到有些意识模糊了，脸上的各种腺体也不受控制，眼泪鼻涕口水，从这台失能的人体逃窜出来。无法自控的不体面让我感到巨大的羞耻。我用还能稍微动作的右手抓紧同样掉在地上的外套，示意央求同事遮住我的脸。

后来 120 急救车来了，另一个产品和 HR 跟车我送去了浙一。医生判断是呼吸性碱中毒，血压比较低，给我开了点药，让我注意休息。

和我自己的判断一样，获得一条难以执行的医嘱。

确实是不大的毛病，休息了三五天也就好了。又继续回来上班。只是我当时从椅子上摔下来时膝盖磕破的伤，又晚了两三周才好。

回忆起来，种种画面如在昨日，情绪仍有起伏。

坦诚说，我属于从小受到大东亚儒家文化共荣圈熏陶和洗礼，内心比较认同「学成文武艺，货与帝王家」「提携玉龙为君死」「苦其心志，饿其体肤」那一套的。很多时候，下意识会把对自身的忽略和对理想的追求，当做是一种高价值的道德标准。

不过，在钉钉的这一系列经历，让我更加重视前置的判断，也即，这究竟是不是一件值得的事情？我的智力水平和道德情操是不是该贡献在更值得的地方？我是不是该把对健康的重视当做一种对资源的保护？

保护身体，保持健康，与诸君共勉。

消失的，和不会消失的职业

职业的本质：以作家为例

无招曾经在尖角班（钉钉的百阿）分享过一个观点，就是“以后没有人想当作家了”。这和现在钉钉宣传不必写文档可能是一脉相承的（不必写文档，众人讨论而由 AI 记录总结，这个我认为是提效的，没有问题的，只是二者间可能有关联关系）。

但我认为，写作是一个永远不会消失的职业，因为它本质是一种自我的创造、记录和表达。

一个人说写作作家这个职业会消失，是因为他眼中的写作就是工作，工作中的记录和总结性质的文档和方案文档，甚至他想象文学写作也会消失。但是文学写作不会真正的消失，因为当你真正的把心放在写作上，你就会发现有的人的写作是思维实验，比如博尔赫斯《巴比伦彩票》，刘慈欣《三体》。有的人的写作是感受的传达，强烈的情感的外化，比如邱妙津《蒙马特遗书》，谷崎润一郎《春琴抄》，有的人的写作是观点的宣告，是政治的呼吁和号召，比如萨特《他人即地狱》，恩格斯《英国工人阶级现状》。 写作、文字，只是一个媒介，一个思想进入公共区域的媒介，只要前者不消失，这个行当就永远不会失去。

海德格尔在《什么叫思想？》里问过一个今天格外刺耳的问题：为什么人人都在发表意见，人人都在处理信息，人人都在使用理性工具，我们却依然未必会思考？

AI 会让很多文字来得更快，也会让很多人更容易误以为自己已经想过。

角色模糊化

受 AI 这团火烤，职业和岗位角色边界正日益消融。

由于AI coding 非常且越来越新手友好，过去的几个月出现很多名人跨界开发的例子。比如《生化危机》的女主 Milla Jovovich，跨界开源的 AI 记忆系统 MemPalace，比如胡彦斌开发了一个粉丝社区 app「彦火」。

又比如，作为投资机构的红杉中国，于 2025 年 5 月推出的 AI 基准测试工具 xbench。这是首个由投资机构主导的评测体系，用以评估 AI 模型理论能力上限与实际落地价值。

这些变革发生在各行各业。

普华永道最新报告指出，AI 正在催生 「角色收敛(role convergence) 」，原本分散于多个专业岗位的职责正整合为更少、更宽泛的角色，个体开始跨越传统领域工作。微软研究院基于 20 万条 Bing Copilot 交互数据的研究也印证，生成式 AI 正推动岗位概念淡化，任务颗粒化成为主流，人机混合协作重构工作流程。

当 AI 智能体能够接管数据分析、代码编写、文档处理等标准化工作，对人类角色的要求，必然从执行层升维至核心能力掌控层，也就是我们经常说的「下判断」。

还是和「设计」章一样，脱离现象看本质，商业公司最核心可以成立的点，其实只有四条：

1. 把东西做出来；
2. 把东西卖出去；
3. 凝聚团队，降低沟通和开发成本；
4. 维稳，保证商业安全。

故而，团队也极可能简化为四种核心角色：

1. 开发者：融合产品/设计/开发等岗，负责产品交付；
2. 销售：黑销售白销售，把产品卖出的去的就是好销售；
3. 凝聚剂：明确团队目标，协调人类合作；
4. 风控：包括但不限于测试/网安等偏开发安全方面的风控，和法务/财务/公关等公司运行方面的风控。

https://www.pwc.com/us/en/services/consulting/human-resources/role-convergence-ai-workforce-redesign.html

一些同事们的小传

第一个 leader

我来钉钉，有 20%的原因在于无招之前的面试聊得比较愉快。这部分的愉快绝大多数来自于他，我譬如朝露的第一个 +1 。

「发心」章提过，我有很重的学徒心态。在来到钉钉的那个节点，我自认为有一些审美，但苦手于总感觉还不能很好地把眼睛、舌头上的审美，转化到手上，真正成为创造力和生产力。他给我的感受是，他有这样的手，以及他不吝让我从他身上好好学到这部分。

可惜，入职后两周，他拉了我一个 C6 楼下的 M Stand 的日程，告诉我他打算休息一段时间。

我忍不住直言相问：既然已萌退意，那为什么一个月前招我进来的时候，还好言陈情，劝勉我来钉钉更上一层楼呢？他笑了，说自己在钉钉九年多，已经过了「那个阶段」了；现在有其他的人生阶段要走。旋而又说，现在正是钉钉的好时候，也是我的好时候。

我们聊了快一个小时，更多是他对我的谆谆留言，指引我无招需要什么，以及钉钉可能需要什么。

一方面，他对我、对钉钉，都抱有很深的善意；另一方面，他当时说的很多内容，虽点得不深，但都一一验证，证明他确实了解无招，了解钉钉。

一个典型守序善良的人决心出走，究竟是因为什么?

马老师说一个人要离职，要不然是钱少了，要不然是委屈了。他的情况应和这条规律吗?

实际负责 ONE 最久的产品负责人

他是一个温和的、友善的、负责的、换位思考的、会跳街舞的产品负责人。如果不是他，我可能更早就会离开钉钉，但他走得比我早。

我晕倒送 120 的那一次之后，他私下跟我说，你该休息就休息，这不值得，要是我女儿.....

他有两个女儿，所以他身上也有一种明显的女儿父亲的特质。想来他的温和友善，换位思考，与这个身份是正相关的。他很爱家人。手机锁屏的屏保和工位桌上都摆着和女儿的合照，很多回，我看到他在无招办公室门口的等待区等待汇报时，会看着屏保出神。每逢难得的放假，他会带着女儿们出去爬山、跑马拉松。

更早之前还有一回，就是我痛苦转正答辩的时候，是他好言苦劝我把 V6 1000 分的任务做了，也是他，那天产品评完需求已经是 11 点了，又拉了我一个 11 点半的会，和我一起过了一遍我的转正述职材料。

其实我心中已经一万次意气用事，想要不干算了。天生我材必有用，大不了明朝散发弄扁舟，何苦囚困钉钉，受这等细碎折磨。可看他 12 点多和我聊完，又继续一个人回到项目室，准备明天要和开发评审的产品方案，和汇报的文档材料，我又觉得自己该死，不够刻苦，于是诸如「天将降大任于是人也」的词权重，又在我心目中上升了。

敢于快速下判断的产品

我和他是同一天进入 ONE，不同的是，我是新入职钉钉，而他是从其他项目转进来的。

他很好的继承了广东血统的显性表征，夏天日常穿着凉拖鞋在项目室里走来走去；不过冬天则一改夏天的随便，常常穿着高领白色毛衣，浅卡其色马甲，和呢子的 trench coat。

他静态的表情有点像 grumpy cat。性格也像，对人事物容忍度低，缺乏耐心，总是皱着眉快刀斩乱麻，开 PRD 评审会像小台风过境。

和他相反，我之前是那种愿意为精确表达付出复杂度的人，说得难听点就是优柔寡断，拖泥带水。认识他之后，引入了一些他的特征，感觉自己快进化成钮祜禄幽素了。不错。

秉笔设计

他是项目室里另外一只 grumpy cat。和上一只 cat 还曾经相视炸毛，彼此挠过一顿。

我感觉他心里给自己的定位是「曾因酒醉鞭名马」的潇洒尘世游侠客，或者常年超负荷赶稿、对助手画稿精度、分镜、线条容错率极低的管教严格的漫画家（比如手冢治虫），但是总是因为一些现实原因，以一种更幽默的方式出场。比如我见过他直接转发其他人给他报的bug，在群里指责某个设计问题，我们排查一通发现那不是bug，用户的需求有些不合理，他倒也不会嘴硬，就是讪笑挠头。

开始我不是特别喜欢他的性格和表达，劲劲的，欠欠的，但他确实对设计有追求。我原本希望可以从我第一任 +1 那里了解和学习到的 pick up 设计的能力，在他身上 pick 到了不少。

他做过丑得人睡不着的设计。之前提过，ONE 有一个阶段，下面那一大排没有用的 Agent 头像就是他做的。但也做过非常好看的，比如一些框架上的改版，不过有的还没发版上线，我就不在这里剧透了。

他也很喜欢熬夜，做丑设计都熬夜做。最近也听他偶尔说过两回感觉胸闷。

屋子里有甲醛味都知道要先搬出去住住呢，我一向劝人不要在不健康的环境里浸淫太久。

「钉钉不培养」设计

她是一个非常可爱的女孩子，是一朵云，一朵雏菊，一朵小泡芙。

她太适合「朵」这个量词了。

她的眼睛和卧蚕都很大，笑起来非常明显，之前是长头发，几个月前剪短了，卷卷的垂在耳畔。她有一种南法的气质，我最初以为她是地中海那一带留过学回来的，了解后才知道是北美。

她是 25 年的校招。我和她短暂合作过两周，她就被调走了。我不能说她完美做出来我的需求，合作的时间太短，这也不合理。但她让我记住的是，不确定的内容总会主动沟通提问，设计上也努力尽心，分类标注细节全面。她也是我在「秩序」章提到，由于 Scrum 长短期打分制度的不合理而苦受牵连的同学。细想起来，所有的 Scrum里，几乎没有设计做 SM 的。欸，钉钉真是不培养设计，对校招和实习的设计同学不太友好。

前几天我问她怎么样，她说：「别问了，再问我又要哭了。」眼圈红红的，却还只是笑。

欸，我觉得自己已经是没有什么攻击性的人了，她比我的攻击性还弱。和她更加熟悉之后我意识到，可能主要是她来自一个温和健康充满爱意的生长环境。

第一份校招进来的工作，真是给她上了一课。

（还有很多，太多写不完，来不及先不写了）

尾声：做泰坦尼克号的水手

前段时间有个同事找我 coffee chat，说感觉自己在过去的几个月做了非常多、非常杂工作，而且都和自己的本职不太相关，感觉简历花了，下份工作不太好找，比较焦虑。我就安慰她说：

首先，职业和 JD 是对经验的归纳，描述的是求职市场上企业在段时间内认为自己需要的员工画像；需求本来就是动态的。我们在经验上发现，这些职业从求学到求职能稳定不变，很有可能只是一种「火鸡悖论」。

> **火鸡悖论（Turkey Paradox）：也被称为「罗素的火鸡」，是一个用来揭示归纳法的局限性**和**认知盲区**的经典思想实验。火鸡根据 300 天被准时喂养的「大数据」，归纳出「饲养员爱我」的绝对真理。然而感恩节前夕，饲养员带来的不是食物，而是屠刀。

> 一句话概括：过去经验的持续正确，无法预测未来的系统性毁灭。

其次，泰坦尼克号的沉没，不影响船上的水手找到下一份工作。因为：

1. 水手能够上船，说明上船前的能力得到了船长的检验和认可；
2. 水手船上做过切实的工作，船的沉没也和这个部分的工作不一定紧密相关；
3. 沉船经验世界上多少人能有？——何况塌这么大一艘船。这是多么宝贵的经验。

ONE 的探索是一场对 AI 和产品理想的激烈着迷。和所有激情和感性和自我出发的AI 产品一样，高方差的预期让人在痛苦中饱含着希望。

不过我不觉得卡片的探索就盖棺定论失败了，只是时机不对， 预期偏差。比如今年 3 月，红杉中国投资了 Boxy，一款同样基于卡片形式的 proactive agency。

至于 ONE 的结局，我只能说，「这是尘寰中消长数应当，何必枉悲伤。」回归「人是目的还是手段」的话题，如果把 one 看作目的，人看作手段，那么结果当然是有些不妙的。但如果反之，ONE 培养了这个项目里所有的产品设计开发，相信大家都有所感悟，并把这份感悟带到了下一个项目里、下一份工作里。说到这里，为什么大家都走了呢？最近几个月我常常想及这个问题，想来想去也只有一句「人之贤不肖譬如鼠矣，在所自处耳。」

最后，为什么非要写这篇复盘呢？我只是惶恐，如果无人记录，许多事情就会乘着「调整」「收缩」「组织优化」「方向变化」这些轻飘飘的词汇飞走消失。可那些事真正发生过的，是一个个具体的人、具体的房间、具体的夜晚、具体的会议、具体的沉默、具体的荒诞和具体的学到啊。

我无法不写。太多思绪和文字，像胃里的蝴蝶，我一仰头，他们就忍不住从食管中争先恐后地飞涌出来，抛下云雾一样的翅粉，火山灰一样悠悠落回我的脸上，遮住我的视线。于是我才终于可以瞑目，安心地开启我的下一段旅程。

附录：《The Mom Test》精华摘要

"Ounce for ounce, there's no better way to learn what customers want and will buy than this wonderful little book. If you want your new product or new business to succeed, start here."

— **John Mullins**
Author of *The New Business Road Test*



Rob Fitzpatrick

THE MOM TEST

How to talk to customers and learn if your business is a good idea when *everyone is lying to you*

核心思想：不要问别人"觉得你的想法怎么样"，而要问他们"生活中实际怎么做的"。

人们会对你的想法撒谎——出于礼貌、善意、不想伤害你。但他们不会对自己过去的行为撒谎。这本书的核心就是教你如何设计问题，让对方无法给你虚假的鼓励，从而获得真实的市场信号。

三条黄金法则

1. **聊他们的生活，而非你的想法** ——不要推销，要挖掘真实痛点
2. **问过去的具体事实，而非未来的假设** ——"你上次怎么做的？" > "你会不会用？"
3. **少说多听，让对方多讲** ——你说得越多，学到的越少

五个关键概念

1. 恭维是毒药（Compliments are Poison）

当有人说"哇，这个想法太棒了！"——你什么都没学到。恭维让你感觉良好，但信息量为零。它是最危险的反馈形式，因为它让你误以为自己在正确的方向上，实际上对方只是在表达礼貌。

把恭维当作警报信号。 一旦听到赞美，立刻转回具体问题："谢谢，那你现在具体是怎么处理这件事的？"

2. 挖掘承诺和推进（Commitment & Advancement）

对话结束时，如果对方没有给出**真实承诺**，这次对话就是失败的。所谓真实承诺，是对方愿意拿出自己的时间、金钱或声誉来支持你：

- 💰 愿意预付费
- 📅 愿意安排时间参加试用
- 🤝 愿意把你介绍给决策者
- 📝 愿意签意向书

嘴上说"好棒"但不采取任何行动 = 毫无价值。**没有承诺的对话就是失败的对话。** 模糊的"不错"是你能得到的最差结果，因为它既没有验证也没有否定，只是浪费了你的时间。

3. 寻找真相的三层挖掘法

客户反馈有三个层次，你需要一层层往下挖：

- **表层**：他们说想要什么（最不可靠——人们不擅长预测自己的行为）
- **中层**：他们为什么想要（有价值——揭示动机和痛点）
- **深层**：他们目前实际怎么做、花了多少钱和时间（黄金信息——行为不会说谎）

关键追问句（建议背下来）：

4. 避免三种致命错误

错误	表现	后果
钓鱼式提问	引导对方说你想听的话	得到虚假验证，自欺欺人

错误	表现	后果
过早推销	一开口就介绍产品	对方切换到"礼貌模式"，不再说真话
接受模糊回答	对方说"这挺有意思的"你就满足了	你以为验证成功，其实啥也没验证

这三个错误的共同点是：它们都让你感觉很好，但实际上让你离真相更远。

5. 对话的正确结构

每次客户对话应该遵循这个流程：

```
开场 → 提出宏观问题 → 深入具体场景 → 追问行为细节 → 请求承诺/推进 → 结束
```

每次对话结束，你应该带走三样东西：

- ✅ 至少一个你之前不知道的事实
- ✅ 对方的一个具体承诺（或明确拒绝——拒绝也是有价值的信号）
- ✅ 清晰的下一步行动

实操要点

1. **找对人**

   不要随便找人聊。你需要找到**真正有这个痛点的人**，而且最好是**有决策权和预算的人**。早期客户不等于所有人——你要找的是那群痛得最厉害、最迫切需要解决方案的人（Rob 称之为 Early Evangelists）。这群人愿意容忍你产品的不完美，因为他们的问题足够痛。

2. **对话场景设置**

- **非正式场合**效果最好：咖啡、会议间隙、行业活动
- 不要说"我想做个调研"（对方会进入防御模式），而是说"我在学习这个领域，想请教你"
- 对话控制在 **15-20 分钟**，短而高效

1. **记录与整理**

- 对话后 **5 分钟内**记下关键笔记——记忆衰减得比你想象的快
- 和团队分享笔记，不要一个人消化——避免个人偏见过滤信息
- 用事实说话，不要用"我感觉他们喜欢"这种模糊表述

十句精髓

1. *观点毫无价值，行为才有价值。*
2. *人们知道自己的问题，但不知道解决方案。*
3. *如果他们还没有尝试解决这个问题，那这个问题就不够痛。*

4. *"我一定会买"是世界上最不值钱的承诺。*
5. *好的客户对话让你感到害怕——因为你可能听到不想听的真相。*
6. *如果你害怕问某个问题，那这个问题可能最重要。*
7. *任何人都能在 10 分钟内学会 Mom Test，但需要一辈子来养成习惯。*
8. *你不是在卖东西，你是在学习。*
9. *每次对话要么推进（得到承诺），要么失败（得到拒绝）——模糊的"不错"是最差的结果。*
10. *创业失败的最常见原因不是产品做不出来，而是做了没人要的东西。*